# MeGA-MP: Metric Graph Advection Message Passing

**A Physics-Informed Message Passing Operator for Advection-Dominated Metric Graphs**

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Many real-world systems are organized as networks where spatio-temporal dynamics unfold along connections and not discretely between nodes. Examples include utility networks such as water distribution systems or gas networks, electrical grids, and traffic flow networks. Such systems are naturally modeled as *metric graphs*, where edges correspond to one-dimensional Euclidean subspaces connected at vertices. Metric graphs are independent of an underlying global Euclidean space, limiting direct application of typical PINNs and operator-learning methods. Especially transport dynamics like advection require a methodology able to capture antisymmetric and long-range dependencies on graphs, which is itself a challenge. We propose a novel physics-informed message passing operator that encodes linear advection on metric graphs as an inductive bias. In the purely advective setting, the operator provably recovers the exact dynamics up to a theoretically derived discretization error without any training. Combined with trainable components like MLPs, our message passing operator extends to realistic advection-reaction dynamics in water distribution systems, where we achieve superior performance compared to baselines and zero-shot generalization across different graph topologies.

## 1 Introduction

In the last years, physics-informed graph neural networks (GNNs) have shown impressive zero-shot generalization capabilities when addressing dynamical systems on conventional graphs (Thangamuthu et al., 2022). However, many real-world systems do not exhibit discrete node-to-node dynamics. Instead, dynamics unfold along connections between nodes. Examples include quantum dynamics, water distribution systems, gas networks, electrical grids, and traffic flow networks (Böttcher & Porter, 2024; Rossman et al., 2020). While a common approach is to discretize the domain and recover a conventional graph, *metric graphs* naturally capture the geometry of these systems. In a metric graph, each edge is associated with a one-dimensional domain – an interval of edge-dependent length – governed by edge dynamics, expressed as an ordinary differential equation (ODE) or, more commonly, a partial differential equation (PDE). In combination with suitable coupling conditions at nodes, such as continuity or conservation of mass or flux, this defines a coupled system of PDEs (Böttcher & Porter, 2024; Blechschmidt et al., 2025).

We identify *advection*, i.e., the transport of a substance through the presence of a flow field, as the main driver for common edge dynamics on a variety of application domains, such as utility networks like water distribution systems (Rossman et al., 2020) and gas networks (Gugat & Herty, 2022), traffic flow networks (Gugat et al., 2005), telecommunication networks, and blood flow through vascular systems (Bressan et al., 2014). From the machine learning (ML) perspective, an intuitive approach is to model such dynamics with GNNs. However, spatio-temporal advection-dominated dynamics on metric graphs pose well-known challenges in the context of GNNs. First, the directional nature of transport requires antisymmetric message passing. Second, spatial dependencies can become global over time, requiring long-range spatial integration. Furthermore, sharp gradients and shocks are preserved and propagated rather than smoothed out as in diffusion-dominated settings. These high-frequency signals cannot be captured by GNNs with inherent over-smoothing characteristics.

In this work, we make use of a suitable inductive bias for linear advection to overcome these challenges and model the dynamics accurately on node level. More precisely, we avoid computationally expensive sampling of function evaluations along each edge space by solving the linear advection PDE using the Method of Characteristics (MoC) and integrating the results in an iterative message passing architecture. This enables an efficient and exact realization of the linear advection dynamics while maintaining end-to-end adaptability if observational data is available.

**Contributions** We propose *metric graph advection message passing (MeGA-MP)*, a message passing neural network (MPNN) framework that provably models linear advection on a metric graph iteratively without requiring learning. More precisely, we rigorously derive MeGA-MP directly from the underlying physical priors and provide a theoretical bound on the discretization error (Section 3). We show how MeGA-MP can be used as a numerical solver for the spatio-temporal node forecasting task on a large-scale and realistic metric graph with advection edge dynamics and demonstrate its overall power by extending it with learning components to solve advection-reaction edge dynamics. Moreover, we investigate the effect of both the purely physics-driven and the learnable components of the model in two ablation studies (Section 4). To benchmark MeGA-MP against classical solvers, we use the 1-dimensional Euclidean domain as a special case of a metric graph (Section 5). A browser-version of our model with live control over inflow conditions is available online[1].

**Related Work** MeGA-MP can serve as a standalone numerical solver for linear advection, expressed as a physics-informed MPNN-architecture. This formulation also allows a straight-forward extension with learnable components into a trainable MPNN to solve more complex, advection-dominated problems on metric graphs. This positions the work at the intersection of numerical PDE solvers for metric graphs, physics-informed ML, and ML for metric graphs. In this order, we will now discuss relevant existing approaches.

Numerical methods for metric graphs largely are specifically designed for quantum graphs, i.e., metric graphs with edge dynamics and coupling conditions associated with the self-adjoint Laplace operator (Arioli & Benzi, 2018; Brio et al., 2022; Böttcher & Porter, 2024). Such works benefit from the rich existing theory on the Laplace operator (Berkolaiko, 2017; Berkolaiko & Kuchment, 2013; Kuchment, 2003; 2005; 2008). In particular, the Laplace operator on a metric graph with continuity and Kirchhoff-Neumann coupling conditions has an orthogonal eigenbasis, which can be utilized in different ways, e.g., by eigenfunction-based spectral solutions (Arioli & Benzi, 2018; Brio et al., 2022; Böttcher & Porter, 2024). Besides quantum graphs, numeric approaches also exist for specific application domains, such as water distribution systems (Rossman et al., 2020; Shang et al., 2023), gas networks (Gugat & Herty, 2022) or traffic networks (Bressan et al., 2014; Gugat et al., 2005). For a broader overview of application domains and related numeric approaches, we refer to Böttcher & Porter (2024). However, numerical approaches typically require discretization of the continuous edge domains, resulting in large numbers of degrees of freedom and increasing computational and memory costs as the spatial resolution grows. In addition, numerical solvers can become computationally demanding when high spatial or temporal resolution is required. These challenges have motivated the development of ML-based approaches for solving dynamical systems.

To this end, Raissi et al. (2019) introduced physics-informed neural networks (PINNs), a framework for neural networks (NNs) that learns the unknown solution of a given PDE on a continuous time and space domain. Since PDE solutions depend on configurations like initial conditions, boundary conditions, and control signals, PINNs typically learn the solution to one such configuration. Neural operators (Kovachki et al., 2023) address this limitation by learning a map from functions that configure a PDE to its solution. Prominent examples include DeepONet by Lu et al. (2019; 2021) and a variation of neural operators by Li et al. (2020a;b;c; 2024). The majority of these methods are developed for Euclidean domains and do not directly transfer to metric graphs.

This issue was recently addressed in different works that apply domain decomposition techniques. This transforms the learning problem into solving dynamics on multiple Euclidean domains – the edge spaces – while constraining correct coupling at nodes. Blechschmidt et al. (2022) use edge-specific PINNs in such a setup to solve drift-diffusion equations on metric graphs. Since PDE parametrizations are – in line with typical PINN

[1] https://anonymous.4open.science/w/tmp-preprint-BB4F/

setups – not a model input, the approach cannot generalize to changing conditions. Laczkó et al. (2025) apply PINNs in a Richardson iterative scheme to solve the time-independent Schrödinger equation on metric graphs. This allows the graph structure to be excluded from model training, making the approach graph agnostic. However, generalization to changing PDE parameters is limited. Blechschmidt et al. (2025) address this by utilizing DeepONets and building a *lego*-like graph-agnostic architecture to solve drift-diffusion equations on metric graphs. However, this framework requires a node-level optimization procedure to determine interface boundary values for each new problem instance, introducing additional computational overhead at graph interfaces. Moreover, all edge dynamics considered in these works are diffusion-dominated and thus accompanied by the aforementioned Kirchhoff-Neumann conditions – a coupling condition fundamentally different to the ones required for advection-dominated edge dynamics.

An orthogonal line of work has shown that the discretization of PDEs in time and space yield powerful physics-motivated MPNNs suitable for conventional tasks on graphs, such as node classification (Chamberlain et al., 2021; Eliasof et al., 2021; Rusch et al., 2022; Choi et al., 2023; Eliasof et al., 2024), but they do not directly address the continuous edge domains characteristic of metric graphs. The reason is the difference in geometry between a conventional graph with discrete node-to-node relations vs. a metric graph with continuous edge spaces. See Appendix A for a detailed discussion, which we recommend reading after the next section.

The existing ML-based methods discussed above either learn solutions for individual PDE instances (PINNs), rely on interface optimization procedures (DeepONet-based approaches), or are designed for conventional graphs without continuous edge domains. In contrast, MeGA-MP directly embeds the numerical structure of linear advection on metric graphs into an MPNN architecture, providing a graph-native framework that can be extended with learnable components while preserving the underlying physical coupling mechanisms.

## 2 Background

The common use case for graphs in dynamical system modeling is as a means of discretizing a continuous Euclidean domain $\Omega \subset \mathbb{R}^n$, with nodes representing spatial locations and edges encoding proximity in $\Omega$. For example, a grid is a graph that approximates a multidimensional Euclidean space, preserves the structure of the continuum, and is commonly used when solving PDEs with numerical schemes (Brandstetter et al., 2022). The same concept has also been adopted to formulate physics-motivated MPNN architectures such as ADR-GNN by Eliasof et al. (2024). Such a setup is not practical in cases where dynamical systems govern the evolution of signals in networks such as water distribution systems, gas networks, electrical grids or traffic flow networks. Instead, *metric graphs* can be employed, as they naturally capture the topology of the original domain. In particular, the edges $e_{uv} \in E$ of a metric graph $\mathcal{G} = (V, E)$ are parametrized by lengths $l_{uv} > 0$ and, potentially, capacities $\alpha_{uv} > 0$, and are associated with one-dimensional Euclidean subdomains $\Omega_{uv} = (0, l_{uv})$ aligned with the edge direction. Here, the graph $\mathcal{G}$ describes the system $\Omega := \sqcup_{e_{uv} \in E} \Omega_{uv}$ of such subdomains coupled at nodes $V$, which correspond to the boundaries $\partial\Omega_{uv} = \{0, l_{uv}\}$ of the edge spaces $\Omega_{uv}$. We visualize the difference between metric graphs and graphs that discretize an underlying continuous space using the example of advection in Figure 10 in Appendix A.

**Dynamical Systems on Metric Graphs** In this work, we focus on metric graphs with associated dynamics that govern the spatio-temporal evolution of signals along the graph. We model this system as a finite, connected, and symmetric directed graph $\mathcal{G} = (V, E)$ together with the following properties which link signals $c_v : \mathbb{R} \to \mathbb{R}_{\geq 0}, t \mapsto c_v(t)$ and $c_{uv} : \mathbb{R} \times \Omega_{uv} \to \mathbb{R}_{\geq 0}, (t, z) \mapsto c_{uv}(t, z)$ over time $\mathbb{R}$, at each node $v \in V$ and along the edge space $\Omega_{uv} = (0, l_{uv})$ of each edge $e_{uv} \in E$, respectively.

1. The *edge dynamics* are usually governed by a PDE that describes the change of the signal $c_{uv}$ over time $\mathbb{R}$ and along the edge space $\Omega_{uv}$ of an edge $e_{uv} \in E$, that is,

$$\frac{\partial c_{uv}(t, z)}{\partial t} = f(t, z, c_{uv}, \nu_{uv}) \tag{1}$$

for all $t \in \mathbb{R}$, $z \in \Omega_{uv} = (0, l_{uv})$ and known functions or coefficients $\nu_{uv}$ that parametrize the edge-wise PDE function $f$.

2. The *coupling conditions* define how signals propagate through nodes in between adjacent edge spaces. Specifically, a node $v \in V$ can be understood as the boundary $z = l_{uv}$ or $z = 0$ of multiple adjacent edge spaces $\Omega_{uv}$ or $\Omega_{vu}$ aligned with the direction of the corresponding incoming or outgoing edge $e_{uv} \in E$ or $e_{vu} \in E$, respectively.

**Advection on Metric Graphs** We model the *edge dynamics*, i.e., Equation (1), as the linear advection PDE

$$\frac{\partial c_{uv}(t,z)}{\partial t} = -\nu_{uv}(t)\ \frac{\partial c_{uv}(t,z)}{\partial z} \tag{2}$$

which describes how the signal $c_{uv}$ is transported in presence of a divergence-free flow field $\nu_{uv} : \mathbb{R} \to \mathbb{R}$ over time $\mathbb{R}$ and along the edge space $\Omega_{uv}$. The flow field additionally satisfies

$$\nu_{vu}(t) = -\nu_{uv}(t) \tag{3}$$

and we use the convention that the sign of a flow velocity $\nu_{uv}(t)$ is positive (negative) if the direction of the corresponding edge $e_{uv} \in E$ aligns (does not align) with the direction of the physical flow, i.e., if the physical flow is from $u \in \mathcal{N}(v)$ to $v \in V$ (from $v \in V$ to $u \in \mathcal{N}(v)$).

The *coupling conditions* for advection-dominated dynamics describe how signals propagate in-between different edge spaces. On the one hand, the *mixing at nodes*

$$c_v(t) = \frac{\sum_{u \in \mathcal{N}_-(v,t)} q_{uv}(t) \cdot c_{uv}(t, l_{uv})}{\sum_{u \in \mathcal{N}_-(v,t)} q_{uv}(t)} \tag{4}$$

describes how signals propagate from edges into nodes. On the other hand, the *inflow continuity condition*

$$c_{uv}(t, 0) = c_u(t) \text{ for all } u \in \mathcal{N}_-(v,t) \tag{5}$$

describes how signals propagate from nodes into edges and can be seen as the *local inflow boundary condition* of the edge dynamics (Eq. (2)). In Equation (4), $q_{uv}(t)$ defines the *flow rate* $q_{uv} = \alpha_{uv}\nu_{uv} : \mathbb{R} \to \mathbb{R}$ from (temporal) *inflow neighbors* $u \in \mathcal{N}_-(v,t) = \{u \in \mathcal{N}(v) \mid \nu_{uv}(t) > 0\}$ of $v \in V$ at time $t \in \mathbb{R}$, that are neighbors $u \in \mathcal{N}(v)$ where at time $t$, the physical flow field is directed from $u$ to $v$. For details, we refer to Appendix A.

*Remark* 2.1 (Global vs. local boundary conditions). Note that Equation (4) and (5) define *local* boundary conditions of the edge dynamics, i.e., Equation (2). In contrast, *global* boundary conditions of the overall dynamical system of the metric graph $\mathcal{G}$ correspond to Dirichlet *boundary nodes* $v \in V_\mathrm{b} \subsetneq V$ for which the signal $c_v(t)$ is known at all times $t \in \mathbb{R}$.

## 3 Method: Metric Graph Advection Message Passing

In this section, we cast the advection edge dynamics (Eq. (2)) and corresponding coupling conditions (Eq. (4) and (5)) into a novel message-passing framework that only requires knowledge of the system states $\mathbf{c} = (c_v)_{v \in V}$ at nodes and the flow field $\boldsymbol{\nu} = (\nu_e)_{e \in E}$ along edges. This is based on the observation that Equation (4) is a special case of message-passing (cf. Appendix B) in the form of

$$\begin{aligned} c_v(t) &= \sum_{u \in \mathcal{N}(v)} w_{uv}(t)\ \phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu}) \ \text{ with} \\ w_{uv}(t) &= \frac{\mathrm{ReLU}(q_{uv}(t))}{\sum_{u \in \mathcal{N}(v)} \mathrm{ReLU}(q_{uv}(t))} \qquad \text{and} \qquad \phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu}) = \ c_{uv}(t, l_{uv}), \end{aligned} \tag{6}$$

where $w_{uv}(t)$ are the normalized weights that select inflows $q_{uv}(t) > 0$ to $v \in V$ only and scale the messages generated by the message function $\phi$. In the current formulation, this requires knowledge of the signal $c_{uv}(t, l_{uv})$ at the outflow boundary of each edge space $\Omega_{uv} = (0, l_{uv})$. However, instead of explicitly solving the dynamics along the edge spaces $\Omega = \sqcup_{e_{uv} \in E} \Omega_{uv}$, we derive a representation of $\phi$ that only relies on signals $\mathbf{c}$ at nodes $V$. This leads to the following general problem we are trying to solve.

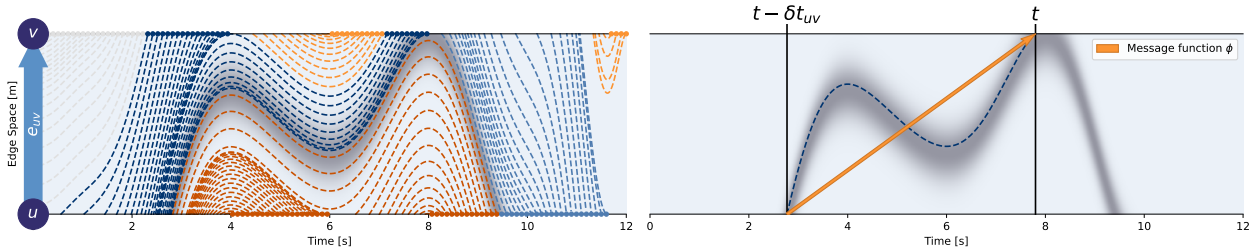



Figure 1: Characteristic curves along the edge $e_{uv} \in E$: If a curve originates at node $u$ at time $t - \delta t_{uv}$ and terminates at node $v$ at time $t$, we call it a *pass-through* (dark blue). If the direction is reversed, we call it *inverse pass-through* (light blue). If a curve starts and ends at the same node, we call it a *self-loop* (dark orange) if that node is $u$ and *inverse self-loop* (light orange) if that node is $v$. The corresponding transport time $\delta t_{uv} > 0$ (right) denotes the duration between the curve's start and end. The shaded area (gray) in the background shows a high-concentration pulse entering the domain at node $u$, that is constant along characteristic curves.

**Definition 3.1** (Spatio-temporal node forecasting with global boundary conditions)**.** Let $V_\mathrm{b} \subsetneq V$ be a subset of global boundary nodes (cf. Remark 2.1) and $T = \{t_i \in \mathbb{R} \mid i \in I = \{1, ..., n\}\}$ a set of discrete times. For a fixed time $t_i \in T$, we define the *history set* $T_\mathrm{hi}(t_i) := \{t_1, ..., t_i\}$ with cardinality $n_\mathrm{hi}(t_i) := |T_\mathrm{hi}(t_i)| = i$ and the *future set* $T_\mathrm{fu}(t_i) := \{t_{i+1}, ..., t_n\}$ with cardinality $n_\mathrm{fu}(t_i) = |T_\mathrm{fu}(t_i)| = n - i$. For simplicity, we omit the dependency of $n_\mathrm{hi}$, $n_\mathrm{fu}$, $T_\mathrm{hi}$ and $T_\mathrm{fu}$ on $t_i$ if $t_i$ is clear from the context. The spatio-temporal node forecasting problem with global boundary conditions refers to the learning problem $\big((c_v(t))_{v \in V \setminus V_\mathrm{b}, t \in T_\mathrm{hi}}, (c_v(t))_{v \in V_\mathrm{b}, t \in T_\mathrm{hi} \sqcup T_\mathrm{fu}}\big) \longmapsto (c_v(t))_{v \in V \setminus V_\mathrm{b}, t \in T_\mathrm{fu}}$, that is the problem of inferring the signal $\mathbf{c}$ at times in the future set from the signal $\mathbf{c}$ at times in the history set and at boundary nodes $V_\mathrm{b}$.

Throughout the rest of this section, we propose MeGA-MP, a MPNN that solves this problem for advection-dominated edge dynamics, where the signal $\mathbf{c}$ usually refers to a **c**oncentration. More specifically, in Section 3.1, we derive advection message passing by relating the *edge dynamics* to *message generation* and the *coupling conditions* to *message aggregation and update*. In Section 3.2, we present the overall MeGA-MP algorithm taking numerical aspects due to the discretization of time (Definition 3.1) into account.

### 3.1 Physical Derivation of MeGA-MP

**Characteristic Curves** We model advection of the concentration $c_{uv} : \mathbb{R} \times \Omega_{uv} \to \mathbb{R}_{\geq 0}$ over time $\mathbb{R}$ and along the edge space $\Omega_{uv} = (0, l_{uv})$ in between two nodes $v \in V$ and $u \in \mathcal{N}(v)$ as the transport process governed by Equation (2) with incompressible flow field $\nu_{uv} : \mathbb{R} \to \mathbb{R}$. In this case, by the Method of Characteristics (MoC), we can derive *characteristic curves* along which the function $c_{uv}$ remains constant over time $\mathbb{R}$ and space $\Omega_{uv}$.

**Lemma 3.2** (Constant concentration along characteristic curves)**.** *Let $e_{uv} \in E$ hold. If the function $c_{uv}$ obeys Equation* (2)*, for each $t_0 \in \mathbb{R}$ and $z_0 \in \mathbb{R}$, the curve*

$$\gamma_{t_0,z_0} : \; T_{t_0,z_0} \longrightarrow \mathbb{R}_{\geq 0}, \quad t \longmapsto c_{uv}\Big(t, z_0 + \textstyle\int_{t_0}^{t} \nu_{uv}(s)\; ds\Big)$$

*is constant.*

An intuition on Lemma 3.2 can be found in Appendix C.1.1. Figure 1 (left) visualizes several characteristic curves for an exemplary flow field $\nu_{uv} : \mathbb{R} \to \mathbb{R}$ along an edge $e_{uv} \in E$ of length $l_{uv} = 1[m]$. Following Lemma 3.2, the concentration $c_{uv}$ is the same at all positions along such a curve. Consecutively, for the example given in Figure 1 (right), knowing the concentration $c_u(t - \delta t_{uv})$ at the node $u \in \mathcal{N}(v)$ $(z = 0)$ is sufficient to determine the concentration $c_v(t)$ at node $v \in V$ $(z = l_{uv})$. This can be observed for the heatmap in the background of the figure, which shows a high-concentration pulse traveling from one boundary at $z = 0$ to the other at $z = l_{uv}$ in $\delta t_{uv} \approx 5[s]$.

**Transport Times** To formulate this relationship as a message function, we need a way to compute the start and end coordinates of characteristic curves that correspond to the sender and receiver node, respectively, as well as the time shift $\delta t_{uv}$, which we call *transport time*. We also use specific terminology to differentiate the following situations, which are also described visually in Figure 1 (left). To get a deeper understanding of the different kinds of transport times, details can also be found in Appendix C.1.1.

**Definition 3.3** (Transport times)**.** Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. If there exists a $\delta t_{uv} := \delta t_{uv}(t) \in \mathbb{R}$, such that

$$\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds = l_{uv} \qquad (-l_{uv}, 0, 0\ ) \text{ and}$$

$$\int_{t-\delta t_{uv}}^{t'} \nu_{uv}(s)\ ds \in (0, l_{uv}) \qquad ((-l_{uv}, 0), (0, l_{uv}), (-l_{uv}, 0))$$

holds for all $t' \in (\min\{t - \delta t_{uv}, t\}, \max\{t - \delta t_{uv}, t\})$, we call $\delta t_{uv}$ a *transport time*, or, more concretely, the *pass-through* (*inverse pass-through*, *self-loop*, *inverse self-loop*) *time with respect to (w.r.t.) the edge* $e_{uv} \in E$ *and time* $t \in \mathbb{R}$. Usually, $e_{uv} \in E$ and $t \in \mathbb{R}$ are clear from the context and we just say *pass-through* (*inverse pass-through*, *self-loop*, *inverse self-loop*) *time*, just as we omit the dependency of $\delta t_{uv}$ on $t$ for better readability.

**Message Function** If the flow $\nu_{uv}$ is constant over time, the transport time $\delta t_{uv}$ is also constant, and concentrations are shifted between the nodes $u$ and $v$. If the flow $\nu_{uv}$ varies, so does the transport time $\delta t_{uv}$, and instead of a clear shift, we observe more complex *time warping*. As this generalizes shift, we use the prior as our message function. To compute a unique positive transport time for a given time $t \in \mathbb{R}$, we need to measure the distance traveled in an arbitrary time interval $[t - \delta t, t]$. For this purpose, we define the function

$$\not{z} : \mathbb{R} \longrightarrow \mathbb{R},\ \delta t \longmapsto \not{z}(\delta t) := \int_{t-\delta t}^{t} \nu_{uv}(s)\ ds, \tag{7}$$

where we omit the dependency of $\not{z}$ on $t \in \mathbb{R}$, $e_{uv} \in E$ and the overall flow field $\boldsymbol{\nu}$ for better readability. Using $\not{z}$, we can compute a positive transport time as the solution to the following optimization problem (OP):

$$\delta t_{uv} := \min \Big\{ \delta t > 0 \ \Big|\ \not{z}(\delta t) \in \{-l_{uv}, 0, l_{uv}\} \Big\}. \tag{8}$$

In Appendix C.1.1, we prove that if there exists a solution, this solution uniquely determines one of the transport times from Definition 3.3 (cf. Theorem C.13). This in turn allows to define our advection-message-generation functions as

$$\phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu}) := \begin{cases} c_u(t - \delta t_{uv}) & \text{if } \not{z}(\delta t_{uv}) = \pm l_{uv} \\ c_v(t - \delta t_{uv}) & \text{if } \not{z}(\delta t_{uv}) = 0 \end{cases}. \tag{9}$$

Especially to mention, using $\not{z}$ again, $\phi$ automatically identifies whether information is carried along a(n) (inverse) pass-through or a(n) (inverse) self-loop and outputs the information from the neighbor $u \in \mathcal{N}(v)$ or the node $v$ itself accordingly. A rigorous derivation of Equation (9) under consideration of physical priors and the intuition behind this can be found in Appendix C.1.1.

**Metric Graph Advection Message Passing** We define the metric graph advection message-passing scheme that determines the concentration $c_v(t)$ at a node $v \in V$ and time $t \in \mathbb{R}$ as

$$c_v(t) = \sum_{u \in \mathcal{N}(v)} w_{uv}(t) \cdot \phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu}) \tag{10}$$

with $w_{uv}(t)$ as defined in Equation (6) and $\phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu})$ as defined in Equation (9).

**Theoretical Guarantees** As elaborated in Section 2, advection on metric graphs is defined by multiple physical priors, the edge dynamics (Eq. (2)) and the coupling conditions (Eq. (4)) and Eq. (5)). While the former are defined on the open edge spaces $\Omega_{uv} = (0, l_{uv})$ only, the latter rely on the values $c_{uv}(\cdot, 0)$ and $c_{uv}(\cdot, l_{uv})$ on the boundary $\partial\Omega_{uv} = \{0, l_{uv}\}$ of $\Omega_{uv}$. Therefore, suitable smoothness assumptions on $c_{uv}$ and $\nu_{uv}$ are required in order to relate these principles.

**Assumption 3.4** (Smoothness assumptions)**.** For each $e_{uv} \in E$, let $c_{uv} : \mathbb{R} \times (0, l_{uv}) \to \mathbb{R}_{\geq 0}$ be differentiable and $c_{uv} : \mathbb{R} \times [0, l_{uv}] \to \mathbb{R}_{\geq 0}$ and $\nu_{uv} : \mathbb{R} \to \mathbb{R}$ continuous.

Under these assumptions, in Appendix C.1, we extend the findings from Lemma 3.2 and investigate the different phenomena as introduced in Definition 3.3 more thoroughly. Based on that, we provide a rigorous theory of why our advection message-passing scheme models the described physics *exactly*, with the main results as follows:

**Theorem 3.5** (Physical alignment)**.** *Let $v \in V$ and $t \in \mathbb{R}$ hold. We assume that Assumption 3.4 holds and that for each $u \in \mathcal{N}(v)$, the OP* (8) *has a solution. If the function $c_v$ obeys the mixing at nodes (Eq.* (4)*) and if for each $u \in \mathcal{N}(v)$, the function $c_{uv}$ obeys the advection edge dynamics (Eq.* (2)*) with inflow continuity condition (Eq.* (5)*), then Equation* (10) *with $\phi$ as defined in Equation* (9) *holds.*

### 3.2 Final Algorithm: MeGA-MP

So far, we have introduced the advection message-passing operator (Eq. (10)). We will now explain how it is applied in practice. For this, two aspects remain to be addressed: First, Equation (10) is defined for continuous time $t \in \mathbb{R}$, whereas in practice boundary values and flow fields are typically known as measurements at discrete points in time. We therefore adapt Equation (10) to handle discrete temporal information, described in paragraph *Temporal Discretization*. Second, the edge lengths $\mathbf{l} = (l_e)_{e \in E}$ and the flow velocities $\boldsymbol{\nu} = (\nu_e)_{e \in E}$ determine how far information travels over the metric graph. Under fast flow conditions, information can traverse multiple edges of the metric graph within the time interval $[t_1, t_n]$ (cf. Definition 3.1 and Appendix D). This cannot be captured by a single application of Equation (10), as it depends only on the immediate 1-hop node neighborhood. We therefore adopt an iterative message-passing scheme, described in paragraph *Model Iterations*. We define the resulting iterative scheme described by Equation (11) as **metric graph advection message passing (MeGA-MP)**.

**Temporal Discretization** We model the concentration $c_v : \mathbb{R} \to \mathbb{R}_{\geq 0}$ at a node $v \in V$ as a discrete time series $(c_v(t))_{t \in T}$ over equidistant discrete times $t \in T = \{t_i \in \mathbb{R} \mid i \in I = \{1, ..., n\}\}$ with sampling interval $dt = t_{i+1} - t_i$. In accordance to Definition 3.1, we aim to estimate the signal $\mathbf{y}_v := (c_v(t))_{t \in T_{\text{fu}}} \in \mathbb{R}^{n_{\text{fu}}}$ at times $t \in T_{\text{fu}}$ and at all non-boundary nodes $v \in V \setminus V_{\text{b}}$. To accommodate known values at times $t \in T_{\text{hi}}$ and unknown values at times $t \in T_{\text{fu}}$, we pad the known history $\mathbf{x}_v := (c_v(t))_{t \in T_{\text{hi}}} \in \mathbb{R}^{n_{\text{hi}}}$ with zeros, yielding $\hat{\mathbf{c}}_v^{(0)} := (\mathbf{x}_v^T, \mathbf{0}^T) \in \mathbb{R}^{n_{\text{hi}} + n_{\text{fu}}}$. Here, the zero entries are placeholders for the predictions $\hat{\mathbf{y}}_v := (\hat{c}_v(t))_{t \in T_{\text{fu}}} \in \mathbb{R}^{n_{\text{fu}}}$ of $\mathbf{y}_v$ (cf. Fig. 2 and Eq. (11) below). Since the transport times $\delta t_{uv} > 0$ are positive by definition (cf. Eq. (8)), applying Equation (10) to $\hat{\mathbf{c}}_v^{(0)}$ for all $v \in V \setminus V_{\text{b}}$ shifts/warps information forward in time, populating the padded entries with the solutions to the node forecasting problem. However, as both edge lengths $l_{uv}$ and flows $\nu_{uv}$ are still continuous, time shifts $t - \delta t_{uv}$ of the message function $\phi$ (Eq. (9)) are also continuous and not constrained to the discrete time points $T$. Therefore, time warping is approximated by interpolating between the two values at times $t_j \in T$ and $t_{j+1} \in T$ for which $t - \delta t_{uv} \in [t_j, t_{j+1})$ holds. We denote the discrete version of $\phi$ by $\varphi$. In our experiments, we use linear interpolation as detailed in Appendix B.1, but depending on the data, different interpolation methods can be considered.

**Model Iterations** As mentioned above, applying Equation (10) one time to $\hat{\mathbf{c}}_v^{(0)}$ for all $v \in V \setminus V_{\text{b}}$ only fully solves the node forecasting problem if all evaluations of the kind $c_u(t - \delta t_{uv})$ in Equation (9) index known concentration values. This is only the case if either $u \in V_{\text{b}}$ is a boundary node, or if $t - \delta t_{uv} < t_i$ lies temporally before the upper bound of the known history $T_{\text{hi}}$. The latter corresponds to a constraint on the flow field $\boldsymbol{\nu}$. However, generalizing this to arbitrary flow fields can be achieved by iterating Equation (10)

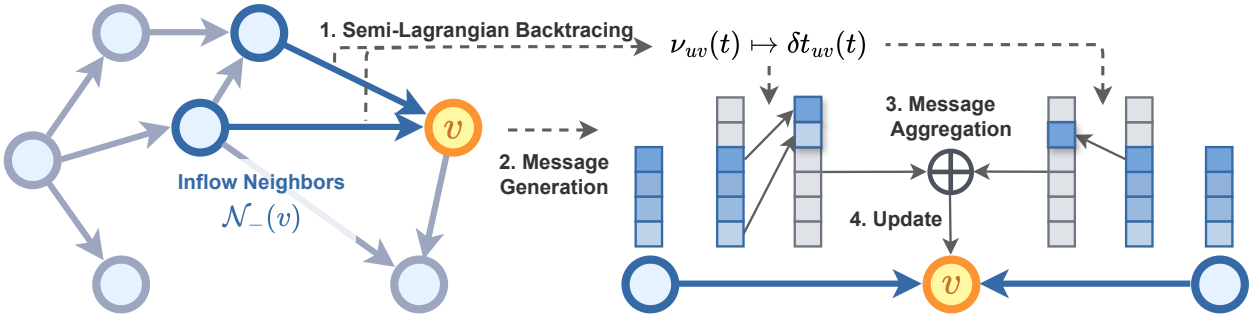



Figure 2: Initial iteration of MeGA-MP (Eq. (11)). Messages are passed from inflow neighbors $u \in \mathcal{N}_-(v,t)$ to the target node $v$ (for all $v \in V$): First, flow velocities $\nu_{uv}$ are converted to the transport time $\delta t_{uv}$ (Eq. (8)). Second, messages are generated via time-warping (Eq. (9)). Third, results are aggregated and assigned to the target node $v$ (Eq. (10)). The graph is directed according to the flow for visualization purposes. Grey entries correspond to padded entries and masked features according to Equation (11).

via

$$
\begin{aligned}
\hat{c}_v^{(0)}(t) &:= \begin{cases} c_v(t) & \text{if } t \in T_{\text{hi}} = \{t_1, ..., t_i\} \\ c_v(t) & \text{if } t \in T_{\text{fu}} = \{t_{i+1}, ..., t_n\} \text{ and } v \in V_{\text{b}} \\ 0 & \text{if } t \in T_{\text{fu}} \text{ and } v \in V \setminus V_{\text{b}} \end{cases}, \\
\hat{c}_v^{(k)}(t) &:= \begin{cases} 0 & \text{if } t \in T_{\text{hi}} \\ 0 & \text{if } t \in T_{\text{fu}} \text{ and } v \in V_{\text{b}} \\ \sum_{u \in \mathcal{N}_-(v,t)} w_{uv}(t)\ \varphi(t, e_{uv}, \hat{\mathbf{c}}^{(k-1)}, \boldsymbol{\nu}) & \text{if } t \in T_{\text{fu}} \text{ and } v \in V \setminus V_{\text{b}} \end{cases} \quad \text{for all } k \geq 1, \\
\hat{c}_v(t) &:= \sum_{k=1}^{\infty} \hat{c}_v^{(k)}(t) \text{ for } t \in T_{\text{fu}},
\end{aligned}
\tag{11}
$$

where $\hat{\mathbf{c}}^{(k-1)} \in \mathbb{R}^{|V| \times (n_{\text{hi}} + n_{\text{fu}})}$ corresponds to the vectorized version of the function $\mathbf{c} = (c_v)_{v \in V}$ that contains the values $\hat{c}_v^{(k-1)}(t)$ for all $v \in V$ and $t \in T = T_{\text{hi}} \cup T_{\text{fu}}$. To build an intuition for how these iterations relax the constraint on $\boldsymbol{\nu}$, note that in a single iteration $k$, a node $v \in V \setminus V_{\text{b}}$ receives information at time $t$ from its neighbors (or itself) at earlier times $t - \delta t_{uv} < t$. This defines a spatio-temporal inflow tree of depth one, along which information is passed. With each iteration, this spatio-temporal tree extends further through the graph. As a result, the node $v$ receives information from increasingly distant nodes at progressively earlier times, eventually from either the known history $T_{\text{hi}}$ or boundary nodes $V_{\text{b}}$. Importantly, only at these leaves, information is initially non-zero. Therefore, summing the values $\hat{c}_v^{(k)}(t)$ over all $k \in \mathbb{N}$ correctly aggregates information from antisymmetrically distributed sending nodes in the inflow tree. A rigorous formalization of this intuition can be found in Appendix C.2. The initial iteration ($k = 0 \mapsto k = 1$) is visualized in Figure 2. Subsequent iterations follow the same schema, but with node feature vectors that already contain information at future time steps $T_{\text{fu}}$.

**Theoretical Guarantees** The following Theorem 3.6 ensures that the prediction $\hat{\mathbf{y}}_v = (\hat{c}_v(t))_{t \in T_{\text{fu}}}$ defined by the last row of Equation (11) is computable in finite time for all $v \in V \setminus V_{\text{b}}$.

**Theorem 3.6** (Convergence of model iterations)**.** *There exists a $\kappa \in \mathbb{N}$, such that $\hat{c}_v^{(k)}(t)$ as defined in Equation (11) is equal to zero for all $k > \kappa$, $v \in V$ and $t \in T_{hi} \cup T_{fu}$.*

Finally, we show that MeGA-MP (Eq. (11)) reproduces the ground-truth (Eq. (9)) if the interpolation error is zero. An extended version of Theorem 3.7 with an error bound obtained through the temporal interpolation can be found in Appendix C.2.

**Theorem 3.7** (Interpolation error)**.** *Let $v \in V$ and $t \in T_{hi} \cup T_{fu}$ hold. Let Assumption C.22 and C.23 hold. If the temporal interpolation error is zero, i.e., if $\phi = \varphi$ holds, we obtain $\hat{c}_v(t) = c_v(t)$.*

*Remark* 3.8 (Initial value problems and efficiency). For the sake of efficiency in both computational and memory complexity, we represent the solution of advection on a metric graph as time series at nodes only (cf. Definition 3.1). The actual values along the 1-dimensional edge spaces are solved implicitly. This enables our model to scale linearly with the number of nodes and, consequently, to handle even large graphs. For this advantage, our method cannot directly receive classical initial value problems (IVPs), where dense values $c_{uv}(t, z)$ along the edge spaces $\Omega_{uv}$ are known at $t = 0$, except when they are constant. However, initial values along edges can be transformed into time series at nodes and vice versa. This is visualized by the gray characteristic curves in Figure 1. We will demonstrate this in Appendix D.2 as an extension of Experiment 2 in Section 5. This can also be used to recover the explicit signal along the edge spaces, as used in the demo.

**Runtime Complexity** The runtime of MeGA-MP depends on edge lengths $l_{uv}$, flow velocities $\nu_{uv}$, number of history and predictions steps $n_{\text{hi}}$ and $n_{\text{fu}}$, respectively, and those of edges $|E|$ in the graph. We analyze runtime in two parts: Computation of transport times and the iterative message passing procedure (cf. Algorithms 1, 2).

The number of iterations required to solve the dynamics depends on the number of edges that information traverses within the future horizon $T_{\text{fu}}$ (cf. Appendix A). An upper bound on the number of iterations can be defined as $k_{\text{up}} = \frac{t_n - t_i}{t_{\min}}$, where $t_{\min} = \min_{e_{uv} \in E, t \in T_{\text{fu}}, \nu_{uv}(t) \neq 0} \frac{l_{uv}}{|\nu_{uv}(t)|}$. In line with other message-passing algorithms, MeGA-MP also applies a message function per edge that processes vectors of length $n = n_{\text{hi}} + n_{\text{fu}}$, resulting in a worst-case runtime of $\mathcal{O}(k_{\text{up}} n |E|)$.

The runtime for computing the transport times $\delta t_{uv}$ from the flow field $\nu_{uv}$ (cf. Algorithm 1) depends on edge lengths $l_{uv}$ and velocities $\nu_{uv}$. The worst case number of iterations of the inner loop is $m = \min(n, \max_{e_{uv} \in E, t \in T_{\text{fu}}, \nu_{uv}(t) \neq 0} \frac{l_{uv}}{|\nu_{uv}(t)| dt})$, and since the outer loop iterates $n$ times for every edge in $E$, the resulting complexity is $\mathcal{O}(|E| m n)$ with an upper bound of $\mathcal{O}(|E| n^2)$. Therefore, the overall worst-case runtime amounts to $\mathcal{O}(|E|(k_{\text{up}} n + n^2))$.

### 3.3 Extended Algorithm: MeGA-MP$^+$

We defined MeGA-MP as the algorithm introduced in Section 3.2, characterized by the message aggregation scheme (Eq. (10)), induced from the mixing at nodes (Eq. (4)), and the physics-informed advection message function $\phi$ (Eq. (9)) within, derived from the advection edge dynamics (Eq. (2)) with inflow continuity condition (Eq. (5)). Formulating MeGA-MP as a message-passing operator allows straight-forward adaptations to more flexible and also trainable models. This can be achieved by extending the (discrete) message-function $\varphi$ (Eq. (11)/(17)). We refer to such extensions of $\varphi$ as $\varphi^+$, and denote the overall extended method as MeGA-MP$^+$. Note that this name stands for a concept rather than a specific method, since depending on the specific formulation of $\varphi^+$, multiple variations are possible. On the one hand, added components can depend on the local metric graph neighborhood $\mathcal{N}(v)$, just as $\varphi$ does. The resulting MeGA-MP$^+$ operator can adapt to spatial dynamics other than linear advection. On the other hand, added components that do not depend on the neighborhood yield a MeGA-MP$^+$ operator that can adapt to additional non-spatial dynamics, while spatially still representing linear-advection. We showcase one such extension in Section 4.

## 4 Experiment 1: Zero-Shot Transfer Learning of Advection-Reaction Dynamics on Real-World Metric Graphs

In this section, we demonstrate the power of MeGA-MP using water quality data, a classical example of an advection-reaction metric graph. To demonstrate that MeGA-MP can be used as a standalone solver for linear advection, but also that it can be extended by a learnable component to account for additional dynamics, we consider two physical configurations: a pure linear advection system, and a coupled advection-reaction system. The objective of this experiment is to solve the node-level regression task, predicting the temporal evolution of signals at nodes, as outlined in Definition 3.1.

We benchmark both MeGA-MP for advection and an example of MeGA-MP$^+$ for advection-reaction dynamics against baseline models, and run an ablation study to show the effect of the different components of both models. Since advection is the only spatial dynamic in both configurations, MeGA-MP and MeGA-MP$^+$

should theoretically generalize to unseen topologies without retraining. We validate this zero-shot capability by testing both configurations on an additional metric graph structure not contained in the training set.

The datasets are described in detail in Section 4.1, the comparison to baselines is in Section 4.2, and the ablation study is in Section 4.3.

### 4.1 Domain and Dataset

**Domain** In this experiment, we consider water distribution systems (WDSs), in which nodes, such as reservoirs or consumer junctions, are connected through pipes. As the pipes are associated with lengths, this domain can be modeled as a metric graph. The WDSs Hanoi (Vrachimis et al., 2018) and L-Town (Vrachimis et al., 2020) are shown in Figures 3 and 4, respectively. Both constitute well-known benchmark networks, used by engineers in the water community.

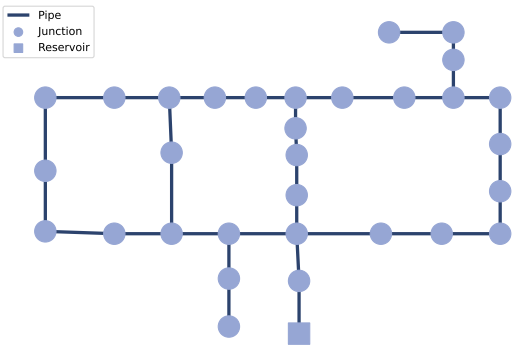



Figure 3: The WDS Hanoi with 32 nodes (31 consumer nodes $V_{\mathrm{c}}$ and one reservoir node $V_{\mathrm{r}}$) and 34 pipes used **for training and testing**.

Figure 4: The WDS L-Town with 784 nodes (782 consumer nodes $V_{\mathrm{c}}$ and two reservoir nodes $V_{\mathrm{r}}$) and 908 pipes used **only for testing**.

A common task in WDSs is to estimate the water quality by predicting the distribution of a substance throughout the WDS. The two main drivers of such processes are advection and *reactions* (Rossman et al., 2020). Reactions describe purely temporal changes in state variables such as chemical and physical transformations, interactions with the environment, and decay. Typically, such dynamics are expressed as ODEs. When combined with advection, they extend Equation (2) as follows:

$$\frac{\partial c_{uv}(t,z)}{\partial t} = -\nu_{uv}(t)\,\frac{\partial c_{uv}(t,z)}{\partial z} + f_r(t,z,e_{uv}), \tag{12}$$

where $f_r$ is a substance-specific reaction term. For this experiment, we simulate the injection of chlorine into the WDS. Chlorine decays over time due to chemical and physical interactions. Details on the specific formalization can be found in Appendix D.1.

**Dataset** We generate two training, validation and test datasets by simulating water quality dynamics on the Hanoi WDS under the two previously mentioned configurations. Specifically, the first dataset models pure linear advection (Eq. (2)), while the second dataset models coupled advection-reaction dynamics (Eq. (12)). To evaluate out-of-distribution topology generalization, we extend each test set by simulating the same dynamics on a different metric graph, the L-Town WDS. All ground truth simulations are generated using EPyt-Flow (Artelt et al., 2024) based on EPANET-MSX (Shang et al., 2023), a hydraulic and water quality simulator for WDSs that supports advection and user-defined reactions such as Equation (32). Details on the parametrization of these setups can be found in Appendix D.1. Each sample of the datasets consists of a time series $T$ over $n = 701$ steps, where the first time step $t_1$ corresponds to the history set $T_{\mathrm{hi}}(t_1) = \{t_1\}$ ($n_{\mathrm{hi}} = 1$) and the remaining $n_{\mathrm{fu}} = 700$ steps correspond to the future set $T_{\mathrm{fu}}(t_1)$ (cf. Definition 3.1).

### 4.2 Experiment 1.1: Comparison to Baselines

**Models** We consider two variants of our method: MeGA-MP, the purely physics-informed operator derived in Section 3.2, and MeGA-MP$^+$, an extension with a non-spatial learnable component enabling adaptation to the advection-reaction dynamics (Eq. (12)) by learning the reaction part from data. In the spirit of Section 3.3, we modify MeGA-MP by extending its message function $\varphi$ (Eq. (11)/(17)) with a multi layer perceptron (MLP). The new message function $\varphi^+$ is designed to be suitable for first-order reaction kinetics, where the solution will be an exponential function. The resulting message function $\varphi^+$ is given by

$$\varphi^+(t, e_{uv}, \hat{\mathbf{c}}^{(k)}, \boldsymbol{\nu}) = \varphi(t, e_{uv}, \hat{\mathbf{c}}^{(k)}, \boldsymbol{\nu}) \cdot \exp(|\delta t_{uv}| \cdot \mathrm{MLP}(\rho_{uv}^{-1})), \tag{13}$$

where $\rho_{uv} \in (0, 1]$ denotes normalized pipe diameters. Details on the architecture and the training of the MLP can be found in Appendix D.1.

To show the ability of our model for transfer learning, we train it on the WDS Hanoi (cf. Figure 3) and test it on both Hanoi and the unseen, much larger WDS L-Town (cf. Figure 4).

**Baselines** As this is the first approach for modeling advection on metric graphs, we select a few representative baselines from the general graph domain. We compare ourselves to the GNN by Gilmer et al. (2017) ("NNConv") as a representative of conventional GNNs, the GNN specifically designed for WDSs by Hermes et al. (2025) ("PDE-GNN"), the transformer GNN by Rampášek et al. (2022) ("GPSConv") as a representative for GNNs potentially able to handle global graph dependencies, and the dynamic-system based GNN by Gravina et al. (2023) ("A-DGN") as a representative for GNNs potentially able to handle dynamical systems with long-range dependencies on graphs. For details regarding these baselines, we also refer to Appendix D.1. There, we also provide details on why the adaptation of the methods for metric graphs from Blechschmidt et al. (2022; 2025) and Laczkó et al. (2025) were not applicable to our task.

Table 1: MAE between simulator ground truth (EPANET-MSX) and model prediction of the **advection** and **advection-reaction** dynamical systems averaged across samples, nodes and time steps of the test datasets. Hanoi is the graph seen in training, L-Town is the generalization network not seen in training. The computation time for both networks and for each method is reported below (batch size = 1). The *preprocessing* time is the time it takes to convert the flow field to transport times (Algorithm 1). For a fair comparison to the simulator, this time has to be added to the computation time of all ML-driven models.

| | WDS | **Advection** | | **Advection-Reaction** | |
|---|---|---|---|---|---|
| | | Hanoi (train/test graph) $\lvert V\rvert = 32, \lvert E\rvert = 34$ | L-Town (test graph) $\lvert V\rvert = 784, \lvert E\rvert = 908$ | Hanoi (train/test graph) $\lvert V\rvert = 32, \lvert E\rvert = 34$ | L-Town (test graph) $\lvert V\rvert = 784, \lvert E\rvert = 908$ |
| MAE | NNConv | 0.1693±0.1610 | 0.2758±0.2345 | 0.1242±0.1132 | 0.2248±0.2113 |
| | PDE-GNN | 0.0442±0.0770 | 0.1640±0.1909 | 0.0651±0.0808 | 0.4136±0.4206 |
| | GPSConv | 0.1878±0.2911 | 0.7429±2.7174 | 0.1381±0.1137 | 0.7319±1.7254 |
| | A-DGN | 0.0916±0.0885 | 0.2568±0.2225 | 0.1841±0.1310 | 0.2606±0.1280 |
| | A-DGN$_{Dia}$ | 0.4260±0.3247 | 0.2502±0.1612 | 0.4441±0.3340 | 0.1614±0.1060 |
| | MeGA-MP | **0.0015**±0.0076 | **0.0017**±0.0061 | 0.1062±0.1240 | 0.2349±0.2236 |
| | MeGA-MP$^+$ | **0.0015**±0.0076 | 0.0018±0.0061 | **0.0013**±0.0037 | **0.0008**±0.0018 |
| Comp. Time (*sec*) | NNConv | 0.0041±0.0278 | 0.0029±0.0003 | 0.0023±0.0001 | 0.0028±0.0001 |
| | PDE-GNN | 1.8648±0.0168 | 1.9251±0.0162 | 1.8926±0.0231 | 1.8925±0.0170 |
| | GPSConv | 0.0085±0.0004 | 0.0145±0.0005 | 0.0083±0.0001 | 0.0176±0.0085 |
| | A-DGN | 0.0266±0.0004 | 0.0271±0.0004 | 0.0260±0.0004 | 0.0267±0.0001 |
| | A-DGN$_{Dia}$ | 0.0074±0.0001 | 0.0420±0.0003 | 0.0073±0.0001 | 0.0418±0.0004 |
| | MeGA-MP | 0.0447±0.0055 | 0.3964±0.0189 | 0.0449±0.0049 | 0.4063±0.0202 |
| | MeGA-MP$^+$ | 0.0450±0.0050 | 0.4068±0.0194 | 0.0450±0.0050 | 0.4036±0.0173 |
| | *preprocessing* | 0.0920±0.0295 | 1.5156±0.1350 | 0.0992±0.0299 | 1.5184±0.1270 |
| | *Simulator* | 0.6724±0.1567 | 31.6906±2.6501 | 0.7520±0.1642 | 34.8503±2.9193 |

**Results and Analysis** Table 1 reports the average mean absolute error (MAE)[2] for both configurations and for both topologies together with the standard deviation over all samples, nodes and time steps. Compared to all baselines, MeGA-MP and MeGA-MP$^+$ show superior performance in the dynamics they are designed for, while MeGA-MP can – as expected – not accurately model the additional reaction dynamics (cf. Section 4.3). Under zero-shot transfer from Hanoi to L-Town, they show no significant increase in error, whereas the performance of the baseline models degrades. An exception is the A-DGN$_{Dia}$ model, however, this baseline underfits in the training setting, indicating that the improvement in performance may be due to factors other than actual generalization. Table 1 also reports the compute time required to run the models on individual samples (batch-size = 1). Since we use the transport times as input to each model, we report the latter without the time it takes to convert the flow field to transport times and instead disclose it as a separate row. Even when adding the preprocessing time to the compute time, each of the models is faster on the large test network L-Town than the simulator that we use as the ground truth. The NNConv model is the fastest, as it is a relatively simple MPNN. The runtime complexity of MeGA-MP and MeGA-MP$^+$, on the other hand, additionally depend on the edge lengths and flow field, as detailed in Section 3.2. However, while MeGA-MP and MeGA-MP$^+$ are slower than most baselines, it is faster than PDE-GNN, which has second-best performance on the in-distribution setting.

We now compare the results qualitatively, first for the pure advection dataset, then for the advection-reaction dataset.

**Pure Advection Configuration:** In Figure 5, we show the predictions of all models evaluated on the metric graph Hanoi. Note that in this case, MeGA-MP does not have to learn, while the other GNN-baselines do. The PDE-GNN models the time series the most accurate, the error mostly stems from misalignment, indicating a failure to properly model transport times. A-DGN can follow the dynamics but exhibits smoothing and fails to capture the high frequency components of the signal. NNConv and GPSConv clearly underfit the data. In contrast, MeGA-MP is able to accurately model the time series at all visualized nodes. In Figure 6, we show the predictions of all models evaluated on L-Town. It can be observed that none of the baselines can precisely capture the signal at all nodes. PDE-GNN and A-DGN can follow the dynamics on some nodes, like nodes 1, 2, 3, and 8. These are the nodes closest to the injection/boundary nodes (stars), which suggests a failure to capture the long-range spatio-temporal dependencies between more distant nodes. MeGA-MP does not show a clear deviation from the ground truth.

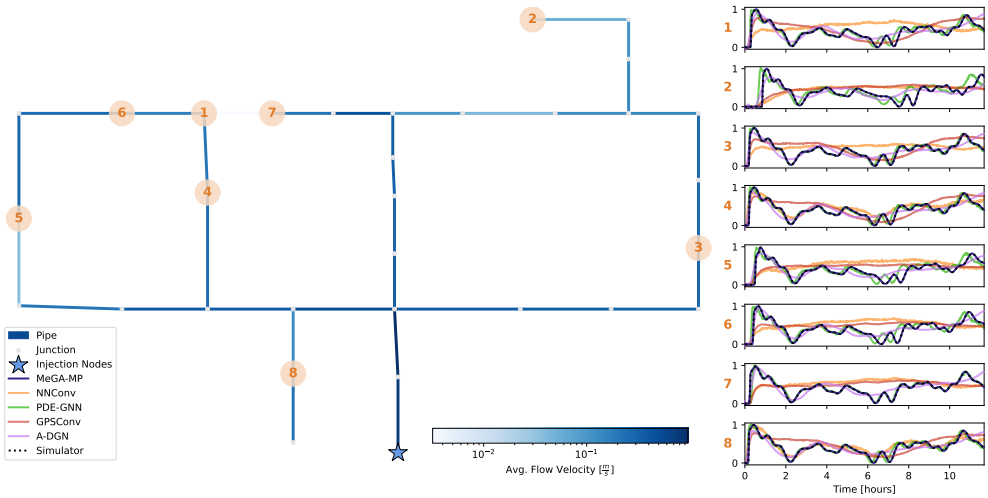



Figure 5: Predictions of baselines and MeGA-MP for the **advection** dynamical system at a randomly chosen subset of nodes of the WDS **Hanoi**. Hanoi is the graph on which all models are trained. Edge color corresponds to the flow velocity averaged across all time steps of the sample.

[2]We clip the predictions of all baselines to the upper and lower bound of chlorine injections as especially PDE-GNN explodes for some samples in the L-Town evaluation dataset.

Figure 6: Predictions of baselines and MeGA-MP for the **advection** dynamical system at a randomly chosen subset of nodes of the WDS **L-Town**. The L-Town metric graph is not part of the training set. Edge color corresponds to the flow velocity averaged across all time steps of the sample.

**Advection-Reaction Configuration:** In Figure 7, we show the predictions of all models evaluated on Hanoi. The observations are similar to the purely advective setting, the only significant difference is that A-DGN underfits here, and for most nodes resembles a highly-smoothed average signal, similar to the other two baselines NN-Conv and GPSConv. Again, MeGA-MP$^+$ accurately models the signal. In Figure 8, showing predictions of all models evaluated zero-shot (without any retraining) on L-Town, a similar degree of underfitting can be observed. Only MeGA-MP$^+$ can model the signal with high precision.

Overall, MeGA-MP and MeGA-MP$^+$ are the only models that perform consistently accurate over multiple nodes and generalize well from the training WDS Hanoi to the test WDS L-Town on the advection and the advection-reaction dynamics, respectively. The different generalization behaviors of the different baselines emphasize the challenges in modeling advection-dominated dynamics on metric graphs and why a suitable physical prior is necessary.

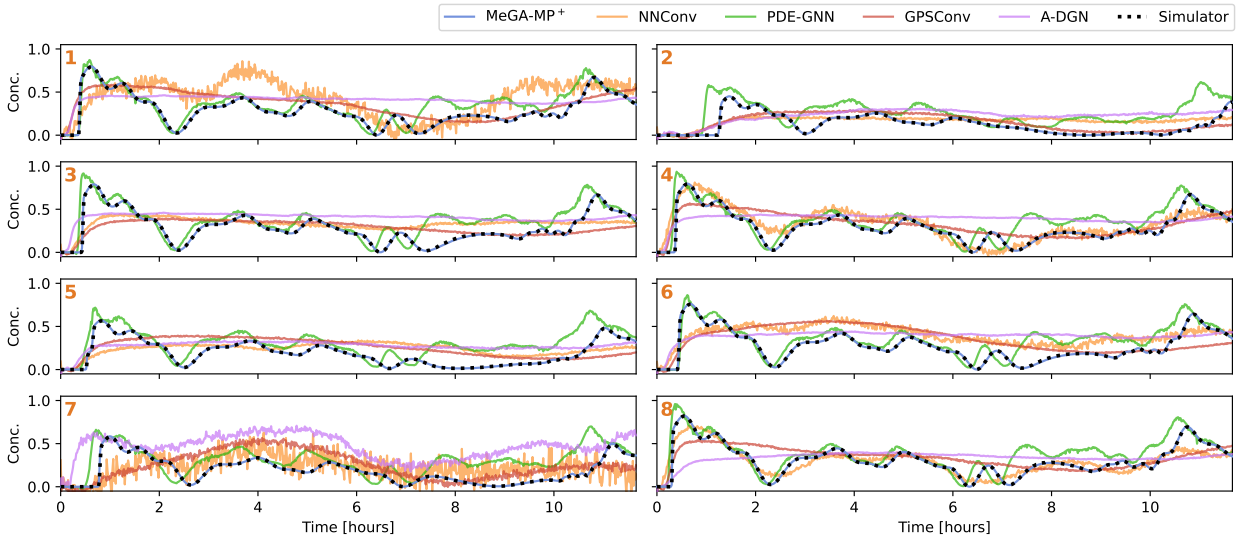



Figure 7: Predictions of baselines and MeGA-MP$^+$ for the **advection-reaction** dynamical system on the WDS **Hanoi**. The selected nodes are the same as in Figure 5.

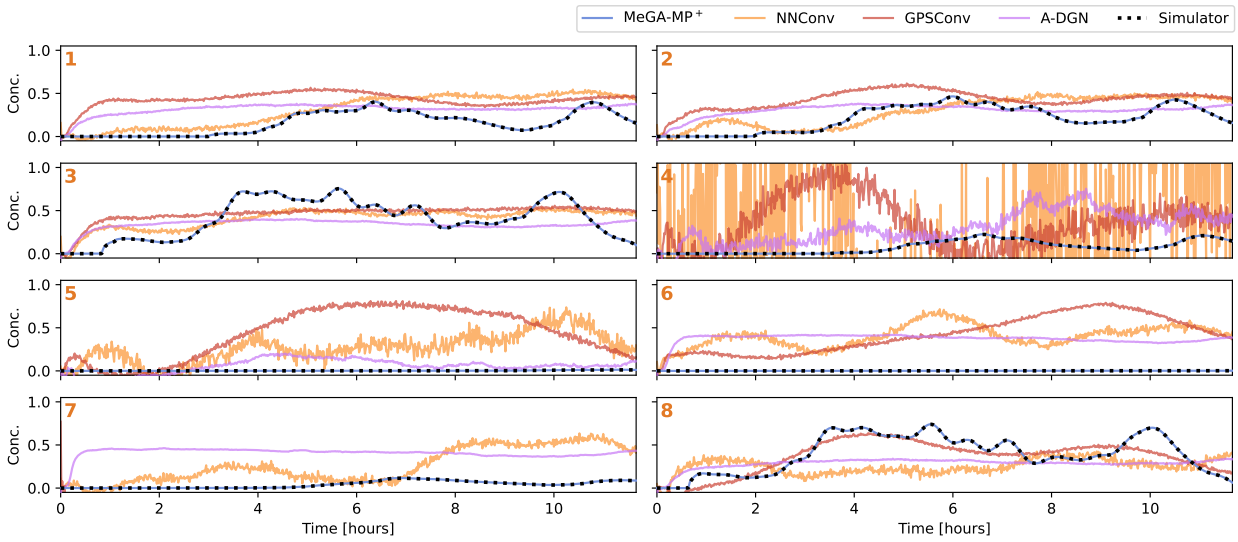



Figure 8: Predictions of baselines and MeGA-MP$^+$ for the **advection-reaction** dynamical system on the WDS **L-Town**. The selected nodes are the same as in Figure 6. In the shown data sample, PDE-GNN explodes and has therefore been excluded.

### 4.3 Experiment 1.2: Ablation Studies

In this section, we investigate how much the physics-informed components of MeGA-MP and MeGA-MP$^+$ contribute to the superior performance observed in Section 4.2.

**Models** In our first ablation study, we investigate the contribution of the two components in the message function $\varphi^+$ of MeGA-MP$^+$, i.e., the non-learnable and physics-informed advection message function $\varphi$ (Eq. (11)/(17)), derived from the advection edge dynamics (Eq. (2)) with inflow continuity condition (Eq. (5)), and the learnable reaction MLP (Eq. (13)). We do so by comparing the performance of MeGA-MP – designed for advection dynamics – and MeGA-MP$^+$ – designed for advection-reaction dynamics – on both the dataset for advection and advection-reaction dynamics.

In our second ablation study, we investigate the contribution of the message aggregation scheme (Eq. (10)), induced from the mixing at nodes (Eq. (4)). We do so by training a model that has the same message aggregation structure as given in Equation (10), but where we replace the message function $\varphi$ not by $\varphi^+$, which yields MeGA-MP$^+$, but by a fully learnable MLP $\varphi^-$. Specifically, we design $\varphi^-$ as a 2-layer MLP with SeLU activation (Klambauer et al., 2017) which has access to adjacent node information over the whole time horizon $T = T_{\text{hi}} \cup T_{\text{fu}}$ and the corresponding pre-processed transport times:

$$\varphi^-(t, e_{uv}, \hat{\mathbf{c}}^{(k-1)}, \boldsymbol{\nu}) = \text{MLP}\left(\hat{\mathbf{c}}_v^{(k-1)}, \hat{\mathbf{c}}_u^{(k-1)}, \delta\mathbf{t}_{uv}\right)_t,$$

where similar to $\hat{\mathbf{c}}^{(k-1)} \in \mathbb{R}^{|V|\times(n_{\text{hi}}+n_{\text{fu}})}$ (cf. Eq. (11)), $\hat{\mathbf{c}}_v^{(k-1)} \in \mathbb{R}^{n_{\text{hi}}+n_{\text{fu}}}$ corresponds to the vectorized version of the function $\mathbf{c} = (c_v)_{v\in V}$ containing the values $\hat{c}_v^{(k-1)}(t)$ for all $t \in T = T_{\text{hi}} \cup T_{\text{fu}}$, and $\delta\mathbf{t}_{uv}$ corresponds to the pre-processed vector $\delta\mathbf{t}_{uv} = (\delta t_{uv})_{t\in T_{hi}\cup T_{fu}} \in \mathbb{R}^{n_{\text{hi}}+n_{\text{fu}}}$ of transport times. We denote this modification of MeGA-MP by MeGA-MP$^-$.

An overview of the different models in these ablations can be found in Table 2. The setup and datasets used in this ablation are equal to the setup of experiment 1 described in Section 4.2.

**Results and Analysis** Table 2 also shows the performance of the different modifications to MeGA-MP. As an extension of Table 2, we provide corresponding figures in Appendix D.1. Firstly, it is evident that MeGA-MP$^-$ performs significantly worse than the other two architectures in the purely advective setting.

Table 2: MAE between simulator ground truth (EPANET-MSX) and predictions of different modifications to MeGA-MP on the **advection** and **advection-reaction** dynamical systems averaged across samples, nodes and time steps of the test datasets. The differences between the models is highlighted by the first three columns.

| | Msg. update via Eq. (10) | Msg. function | Learn. comp. | **Advection** Hanoi (train/test graph) $\lvert V\rvert = 32, \lvert E\rvert = 34$ | **Advection** L-Town (test graph) $\lvert V\rvert = 784, \lvert E\rvert = 908$ | **Advection-Reaction** Hanoi (train/test graph) $\lvert V\rvert = 32, \lvert E\rvert = 34$ | **Advection-Reaction** L-Town (test graph) $\lvert V\rvert = 784, \lvert E\rvert = 908$ |
|---|---|---|---|---|---|---|---|
| MeGA-MP | ✓ | $\varphi$ | × | $\mathbf{0.0015}_{\pm 0.0076}$ | $\mathbf{0.0017}_{\pm 0.0061}$ | $0.1062_{\pm 0.1240}$ | $0.2349_{\pm 0.2236}$ |
| MeGA-MP$^+$ | ✓ | $\varphi^+$ | ✓ | $\mathbf{0.0015}_{\pm 0.0076}$ | $0.0018_{\pm 0.0061}$ | $\mathbf{0.0013}_{\pm 0.0037}$ | $\mathbf{0.0008}_{\pm 0.0018}$ |
| MeGA-MP$^-$ | ✓ | $\varphi^-$ | ✓ | $0.0910_{\pm 0.0875}$ | $0.2169_{\pm 0.1722}$ | $0.0971_{\pm 0.0852}$ | $0.1873_{\pm 0.3045}$ |

This indicates that the full strength of our approach does not only come from the message aggregation scheme (Eq. (10)), induced from the mixing at nodes (Eq. (4), but also from the inductive bias in the advection message functions $\varphi$ (Eq. (11)/(17)) and thus, in $\varphi^+$ (Eq. (13)), derived from the advection edge dynamics (Eq. (2)) with inflow continuity condition (Eq. (5)). Removing the physical priors from $\varphi$ and thus, $\varphi^+$ by replacing it with a fully learnable function $\varphi^-$ does not suffice to model the complex antisymmetric and long-range dependencies induced by the advection dynamics. Secondly, the on-par-performance of MeGA-MP and MeGA-MP$^+$ in the advection setting indicates that the additional MLP is not required when advection is the only dynamic.

In the case of advection-reaction, only MeGA-MP$^+$ is able to accurately capture the added complexity from the reaction term, emphasizing the utility of the simple modification in $\varphi^+$. Due to the inability to learn dynamics, MeGA-MP fails to capture the reaction dynamic without the modification used in MeGA-MP$^+$. Furthermore, the fact that MeGA-MP$^-$ performs worse than MeGA-MP$^+$ indicates that the other GNN-based baselines from Section 4.2 do not perform worse because of a different message aggregation function.

In summary, our ablation studies show the utility of coding the advection characteristics directly into the model architecture. Without it, the model cannot reliably capture the challenging spatio-temporal dependencies. This once again emphasizes the strength of physics-informed ML and carefully constructed inductive bias over purely data-driven approaches in advection-dominated settings.

## 5 Experiment 2: Advection on a 1D Euclidean Domain

In this section, we empirically validate MeGA-MP by using it as a numerical solver for advection in a domain where well-known versatile numerical solvers exist: On a 1-dimensional Euclidean domain. The advantage of this domain is that it also resembles a metric path graph. The goal of this experiment is to compare the characteristics of MeGA-MP to those of classical solvers. To this end, we compare our method against two classical baselines: A semi-Lagrangian solver, as well as a method-of-lines formulation integrating over time with a Runge-Kutta (RK4) solver using a WENO5 scheme (Jiang & Shu, 1996) to approximate the spatial derivatives. For details regarding the baselines, we refer to Appendix D.2.

**Domain and Dataset** The domain for this experiment is $\Omega = [0, l]$, describing a 1-dimensional space of length $l = 100$. We aim to predict the solution $c(t, z)$ of a synthetic boundary value problem (BVP) at equidistant grid points $t$ and $z$ of the time and space interval $[0, 100]$ and $[0, 100]$ for different temporal and spatial discretizations $dt$ and $dz$ from $\{\frac{100}{1000}, \frac{100}{316}, \frac{100}{100}, \frac{100}{32}, \frac{100}{10}\}$, respectively. Conceptually, this models an injection at the boundary $z = 0$, advecting along $\Omega$ over time $t \in [0, 100]$. To enable a fair comparison, we choose the BVP such that an analytical solution exists. The setup is described in detail in Appendix D.2. In light of our discussion on IVPs and efficiency (Remark 3.8), we discuss a similar setup posed as an IVP with boundary values in Appendix D.2, too.

**Results and Analysis** Table 3 shows the MAE of the different solvers at different space-time resolutions. We find that MeGA-MP approximates the ground truth well and provides results competitive to other solvers, as also reflected in Figure 19. It can also be observed that MeGA-MP shows the best performance

Table 3: MAE between analytical ground truth and different solver solutions over all temporal and spatial grid points. The entry "–" indicates divergence of the method. Find the full version of this table and the runtime analysis in Appendix D.2.

| $dt$ | $dz$ | semi- Lagr. | RK4 | MeGA-MP |
|---|---|---|---|---|
| 0.1 | 0.10 | **0.0016**$\pm$0.0050 | – | 0.0024$\pm$0.0081 |
| | 1.00 | 0.0108$\pm$0.0230 | 0.0035$\pm$0.0073 | **0.0012**$\pm$0.0068 |
| | 10.00 | 0.0562$\pm$0.0764 | 0.0288$\pm$0.0415 | **0.0057**$\pm$0.0182 |
| 1.0 | 0.10 | 0.0080$\pm$0.0144 | – | 0.0157$\pm$0.0298 |
| | 1.00 | 0.0097$\pm$0.0137 | **0.0088**$\pm$0.0155 | 0.0103$\pm$0.0208 |
| | 10.00 | 0.0545$\pm$0.0720 | 0.0291$\pm$0.0395 | **0.0115**$\pm$0.0222 |
| 10.0 | 0.10 | **0.0763**$\pm$0.0964 | – | 0.0939$\pm$0.0981 |
| | 1.00 | **0.0728**$\pm$0.0923 | – | 0.0915$\pm$0.0952 |
| | 10.00 | **0.0671**$\pm$0.0842 | 0.0728$\pm$0.0936 | 0.0761$\pm$0.0763 |

when the temporal resolution is high ($dt = 0.1$), while it is less affected by changes in spatial resolution. This is an expected result, as our method interpolates the signal at nodes temporally. Therefore, the interpolation error correlates with the temporal resolution, as derived in depth in Appendix C.2. This is in contrast to the baseline semi-Lagrangian solver, which interpolates spatially and thus benefits from higher spatial resolution. RK4 becomes unstable for certain combinations of spatial and temporal resolutions.

The MeGA-MP solutions for $dz = 1.0$ are visualized in Figure 9, where the numerical smoothing from interpolation is visible and pronounced. The latter is amplified because of the sharp peak and therefore high-frequency content of the signal. For $dt = 10$ one can also observe that smoothing increases in $z$-direction, i.e. in the direction of the flow field, as this coincides with the direction of spatial integration and error accumulation (cf. Appendix C.2). More detailed visualizations of all combinations of space-time discretization, an extended version of Table 3 and a runtime analysis can be found in Appendix D.2.

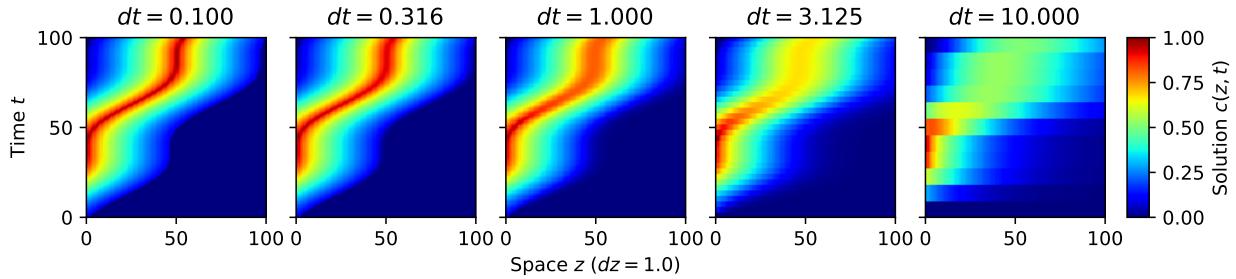



Figure 9: MeGA-MP solution of the BVP (33) for a fixed spatial discretization of $dz = 1.0$ for varying values of temporal discretizations $dt$. Each vertical pixel row corresponds to the time series at a node (i.e., a grid point), each horizontal row corresponds to one time step across all nodes. The sinusoidal flow field (Eq. (35)) is always positive, i.e. the injected signal at the left boundary ($z = 0$) is transported towards the right boundary ($z = 100$).

## 6 Discussion & Conclusion

In this work, we have introduced MeGA-MP, a message-passing formulation solving linear advection dynamical systems on metric graphs. We have shown theoretically and empirically that it is a valid solver with a provable error bound and that it can be extended to solve more challenging dynamics. Here we used an MLP as the extension. However, following the structural archetype of MPNNs in principle also allows more complex integrations with existing GNNs.

The comparison to other PDE solvers shows the saliency of our method, as our solution is comparable to that of established numerical solvers in scenarios where a comparison is possible.

By experimenting on dynamics in WDSs we did not only show that our method can solve advection-dominated dynamics on metric graphs, but also that MeGA-MP is directly applicable as a surrogate model for water quality, specifically the modeling of chlorine. It closely models the simulator EPANET-MSX, while being considerably faster and more flexible, as it integrates much better into the existing ML infrastructure. Successful zero-shot transfer to unseen metric graphs further strengthens the positioning of MeGA-MP as a viable surrogate model. We attribute this transferability to novel geometries mainly to the inductive bias in our message-passing scheme, allowing a separation of spatial integration of graph signals by MeGA-MP and data-driven adaptation to the additional non-spatial reaction dynamic using MeGA-MP$^+$. It is unclear to what extent the transferability persists when learnable spatial components are added. We will discuss this in the following sections, but argue that in the presence of advection, MeGA-MP always contributes a suitable inductive bias at low cost.

**Limitations** MeGA-MP is specifically designed for linear advection-dominated dynamics on metric graphs and is naturally limited to PDEs where linear advection is the dominant spatial dynamic. The approximation error is controlled by two factors: the temporal discretization and the number of message passing iterations required to propagate information across the graph. The latter depends on the edge lengths and flow velocities; short lengths and fast flows require more iterations, leading to greater accumulation of temporal interpolation errors. Both effects are consistent with our theoretical error bound (Theorem 3.7) and empirically supported by Experiment 2 (Section 5). We therefore recommend choosing sufficiently fine temporal discretizations relative to the flow velocities in the system. Beyond numerical accuracy, MeGA-MP currently assumes exact knowledge of the flow field, which could be unavailable in some practical settings. To this end, MeGA-MP does not provide explicit uncertainty quantification, but the low execution time allows feasible Monte-Carlo estimates of uncertainty.

**Future Work** Several directions can build naturally on this work. First, extending MeGA-MP to more complex spatial dynamics, such as semi- and non-linear advection or advection-diffusion, and to more complex temporal dynamics, such as multi-species reactions, via additional learnable operators is a promising avenue. The latter, in particular, would bring this method closer to a full ML-driven surrogate model for the simulator EPANET-MSX for WDSs. Second, exploring inverse problems such as identifying flow fields or reaction coefficients from observational data would not only address the current limitation of requiring exact flow field knowledge, but would further emphasize the practical advantages of such surrogates over numerical simulators. Thirdly, a systematic investigation of robustness to noisy or incomplete parameters such as flow fields is an important prerequisite for deployment of such systems in the real-world. Finally, metric graphs governed by other PDEs beyond advection-dominated edge dynamics pose fundamentally different challenges. A systematic study of these different problem classes, including benchmark datasets and baseline evaluations, is an important direction for future work and the comparability of different methods across domains.

## Impact Statement

This work presents a physics-informed ML method for modeling spatio-temporal dynamics on advection-dominated metric graph domains. The data used in this work is self-generated and does not contain personal or sensitive information, ensuring no privacy concerns under GDPR or comparable regulations, or ethical concerns. However, real-world deployment of such methods may require sensor data or consumption records that can be linked to individual consumers. In such cases, data collection and processing must comply with applicable data protection regulations, and anonymization or aggregation techniques should be applied where possible.

The primary intended application of this work is modeling in scientific and engineering context, with the goal of improving the efficiency and accuracy of simulations. We also acknowledge the potential broader impacts of physics-informed ML including misuse and emphasize that responsible development and independent validation are essential. For example, in safety-critical settings, incorrect model predictions could have

negative consequences for the provision of essential services. Therefore, before deploying such systems in the real world, a thorough investigation of failure modes is essential, including error-proneness under distribution shift, sensitivity to misspecified physical parameters and the absence of uncertainty quantification in the current formulation.

## LLM Statement

This research was partially assisted by using large language models (LLMs) such as ChatGPT, Gemini or Claude. More precisely, LLMs were used for retrieval and discovery, such as finding related work. However, all related work cited in this paper was read and verified by the authors. Moreover, LLMs were used to generate boilerplate code that has been checked for correctness by the authors. Finally, LLMs were used to assist with editing and proofreading of selected parts of the manuscript with a focus on improving clarity and readability. The scientific content, the experimental design, the code implementing the novel methods presented in this work, the results, most parts of the text in this manuscript as well as the theory and equations were not generated by LLMs.

## References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Mario Arioli and Michele Benzi. A Finite Element Method for Quantum Graphs. *IMA Journal of Numerical Analysis*, 38(3):1119–1163, 2018.

André Artelt, Marios S. Kyriakou, Stelios G. Vrachimis, Demetrios G. Eliades, Barbara Hammer, and Marios M. Polycarpou. Epyt-flow: A toolkit for generating water distribution network data. *Journal of Open Source Software*, 9(103):7104, 2024.

Gregory Berkolaiko. An Elementary Introduction to Quantum Graphs. *Geometric and Computational Spectral Theory*, 700:41–72, 2017.

Gregory Berkolaiko and Peter Kuchment. *Introduction to Quantum Graphs*. Number 186. American Mathematical Soc., 2013.

Jan Blechschmidt, Jan-Frederik Pietschman, Tom-Christian Riemer, Martin Stoll, and Max Winkler. A comparison of PINN approaches for drift-diffusion equations on metric graphs. *arXiv preprint arXiv:2205.07195*, 2022.

Jan Blechschmidt, Tom-Christian Riemer, Max Winkler, Martin Stoll, and Jan-F Pietschmann. Physics-Informed DeepONets for drift-diffusion on metric graphs: simulation and parameter identification. In *Proceedings of the 42th International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4482–4506. PMLR, 2025.

Lucas Böttcher and Mason A Porter. Dynamical Processes on Metric Networks. *arXiv preprint arXiv:2401.00735*, 2024.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message Passing Neural PDE Solvers. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.

Alberto Bressan, Sunčica Čanić, Mauro Garavello, Michael Herty, and Benedetto Piccoli. Flows on Networks: Recent Results and Perspectives. *EMS Surveys in Mathematical Sciences*, 1(1):47–111, 2014.

Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *CoRR*, abs/1711.07553, 2017. URL http://arxiv.org/abs/1711.07553.

Moysey Brio, J-G Caputo, and Hannah Kravitz. Spectral Solutions of PDEs on Networks. *Applied Numerical Mathematics*, 172:99–117, 2022.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. GRAND: Graph Neural Diffusion. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1407–1418. PMLR, 2021.

Airlie Chapman. Advection on Graphs. *Semi-Autonomous Networks: Effective Control of Networked Systems through Protocols, Design, and Modeling*, pp. 3–16, 2015.

Jeongwhan Choi, Seoyoung Hong, Noseong Park, and Sung-Bae Cho. GREAD: Graph Neural Reaction-Diffusion Networks. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 5722–5747. PMLR, 2023.

Moshe Eliasof, Eldad Haber, and Eran Treister. PDE-GCN: Novel Architectures for Graph Neural Networks Motivated by Partial Differential Equations. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Moshe Eliasof, Eldad Haber, and Eran Treister. Feature Transportation Improves Graph Neural Networks. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, number 11, pp. 11874–11882, 2024.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural Message Massing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 2017.

Alessio Gravina, Davide Bacciu, and Claudio Gallicchio. Anti-Symmetric DGN: A Stable Architecture for Deep Graph Networks. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

Martin Gugat and Michael Herty. *Modeling, Control, and Numerics of Gas Networks*, volume 23, pp. 59–86. Elsevier, 2022.

Martin Gugat, Michael Herty, Axel Klar, and Günter Leugering. Optimal Control for Traffic Flow Networks. *Journal of Optimization Theory and Applications*, 126(3):589–616, 2005.

Richard Habermann. *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*, volume 5. Pearson, 2013.

Luca Hermes, André Artelt, Stelios G Vrachimis, Marios M Polycarpou, and Barbara Hammer. A Benchmark for Physics-informed Machine Learning of Chlorine Concentration States in Water Distribution Networks. *SN Computer Science*, 6(5):522, 2025.

Weihua Hu*, Bowen Liu*, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJlWWJSFDH.

Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted eno schemes. *Journal of Computational Physics*, 126(1):202–228, 1996. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.1996.0130. URL https://www.sciencedirect.com/science/article/pii/S0021999196901308.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/5d44ee6f2c3f71b73125876103c8f6c4-Paper.pdf.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs. *The Journal of Machine Learning Research*, 24(1):1–97, 2023.

Peter Kuchment. Quantum graphs: I. Some Basic Structures. *Waves in Random media*, 14(1):S107, 2003.

Peter Kuchment. Quantum Graphs: II. Some Spectral Properties of Quantum and Combinatorial Graphs. *Journal of Physics A: Mathematical and General*, 38(22):4887–4900, 2005.

Peter Kuchment. Quantum Graphs: An Introduction and a Brief Survey. *arXiv preprint arXiv:0802.3442*, 2008.

Csongor L Laczkó, Mihály A Vághy, and Mihály Kovács. A transferable PINN-based method for quantum graphs with unseen structure. *IFAC-PapersOnLine*, 59(1):67–72, 2025.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural Operator: Graph Kernel Network for Partial Differential Equations. *arXiv preprint arXiv:2003.03485*, 2020b.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole Graph Neural Operator for Parametric Partial Differential Equations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, 2020c.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-Informed Neural Operator for Learning Partial Differential Equations. *ACM/IMS Journal of Data Science*, 1(3):1–27, 2024.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Ladislav Rampášek, Mikhail Galkin, Vijay Dwivedi, Anh Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Lewis A Rossman, H Woo, M Tryby, F Shang, R Janke, and T Haxton. EPANET 2.2 Users Manual. *United States Environmental Protection Agency (EPA).*, 2020.

T Konstantin Rusch, Ben Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael Bronstein. Graph-Coupled Oscillator Networks. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18888–18909. PMLR, 2022.

Feng Shang, Lewis A Rossman, and James G Uber. EPANET-MSX 2.0 User Manual. *United States Environmental Protection Agency (EPA)*, 2023.

Abishek Thangamuthu, Gunjan Kumar, Suresh Bishnoi, Ravinder Bhattoo, NM Krishnan, and Sayan Ranu. Unravelling the Performance of Physics-informed Graph Neural Networks for Dynamical Systems. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Stelios G. Vrachimis, Marios S. Kyriakou, Demetrios G. Eliades, and Marios M. Polycarpou. LeakDB: A Benchmark Dataset for Leakage Diagnosis in Water Distribution Networks. In *Proceedings of the 1st International Joint Conference on Water Distribution Systems Analysis & Computing and Control for the Water Industry (WDSA/CCWI)*, 2018.

Stelios G. Vrachimis, Demetrios G. Eliades, Riccardo Taormina, Avi Ostfeld, Zoran Kapelan, Shuming Liu, Marios Kyriakou, Pavlos Pavlou, Mengning Qiu, and Marios M. Polycarpou. BattLeDIM: Battle of the Leakage Detection and Isolation Methods. In *Proceedings of the 2nd International Joint Conference on Water Distribution Systems Analysis & Computing and Control for the Water Industry (WDSA/CCWI)*, pp. 1–6, 2020.

Shuhei Watanabe. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance, 2025. URL https://arxiv.org/abs/2304.11127.

## Organization of the Supplementary Material

**A Details on the Background** **23**

**B Details on the Method** **25**

B.1 Final Algorithm: MeGA-MP . . . 25

**C Details on Theoretical Guarantees** **27**

C.1 Physical Derivation of MeGA-MP . . . 27

C.1.1 Message Generation: Advective Transport Along Edges . . . 28

C.1.2 Message Aggregation and Update: Mixing at Nodes . . . 34

C.2 Error Bounds on MeGA-MP . . . 35

C.2.1 Preliminaries: Trees with Children-Aggregation Scheme . . . 35

C.2.2 Construction of Inflow Trees . . . 37

**D Details on the Experiments** **39**

D.1 Experiment 1: Zero-Shot Transfer Learning of Advection-Reaction Dynamics on Real-World Metric Graphs . . . 39

D.2 Experiment 2: Advection on a 1D Euclidean Domain . . . 44

**E Proofs** **46**

E.1 Proof of Lemma 3.2 and C.2 . . . 46

E.2 Proof of Lemma C.4 . . . 49

E.3 Proof of Theorem C.5 . . . 51

E.4 Proof of Lemma C.7 . . . 52

E.5 Proof of Lemma C.8 . . . 54

E.6 Proof of Lemma C.9 . . . 56

E.7 Proof of Lemma C.10 . . . 57

E.8 Proof of Theorem C.11 . . . 57

E.9 Proof of Theorem C.12 . . . 60

E.10 Proof of Theorem C.13 . . . 63

E.11 Proof of Lemma C.15 . . . 66

E.12 Proof of Theorem 3.5 and C.16 . . . 67

E.13 Proof of Theorem C.17 . . . 69

E.14 Proof of Theorem C.20 . . . 71

E.15 Proof of Theorem C.21 . . . 72

E.16 Proof of Theorem 3.6 . . . 74

E.17 Proof of Theorem 3.7 and Theorem C.25 . . . 74

# A Details on the Background

**Advection on Metric Graphs** Generally speaking, advection describes transport through the presence of a flow field $\nu_{uv} : \mathbb{R} \times \Omega_{uv} \to \mathbb{R}$. For each time $t \in \mathbb{R}$, the flow $\nu_{uv}(t, \cdot) : \Omega_{uv} \to \mathbb{R}$ defines a one-dimensional vector field along the edge space $\Omega_{uv} = (0, l_{uv})$. By definition of incompressibility, $\mathrm{div}(\nu_{uv}(t, \cdot)) = \frac{\partial \nu_{uv}(t,\cdot)}{\partial z} = 0$ holds, or equivalently, the vector field $\nu_{uv}(t, \cdot)$ is constant on $\Omega_{uv} = (0, l_{uv})$, modeling the edge $e_{uv} \in E$. Analogously, the vector field $\nu_{vu}(t, \cdot)$ is constant on $\Omega_{vu} = (0, l_{vu}) = (0, l_{uv})$, modeling the contrasting edge $e_{vu} \in E$. Since both constant vector fields $\nu_{uv}(t, \cdot)$ and $\nu_{vu}(t, \cdot)$ model the same physical phenomenon, namely the constant, actually observable *physical* flow field between the nodes $v \in V$ and $u \in \mathcal{N}(v)$, but along contrasting directions (the beginning of the edge $e_{uv}$ is the end of the edge $e_{vu}$ and vice versa), the orientation of $\nu_{vu}(t, \cdot)$ must be the opposite of the one of $\nu_{uv}(t, \cdot)$. Since they both correspond to the same, constant physical flow field, $\nu_{vu}(t, \cdot) = -\nu_{uv}(t, \cdot)$ (Eq. (3)) must hold, and since they do not depend on space, we omit the spatial dependency.

Together with the convention of signs (cf. Section 2), this allows to formally define the inflow neighbors from $v \in V$ at time $t \in \mathbb{R}$, i.e., neighbors where there is a flow from $u \in \mathcal{N}(v)$ *in*to $v \in V$ at time $t \in \mathbb{R}$, as

$$\mathcal{N}_-(v, t) := \{u \in \mathcal{N}(v) \mid \mathrm{sgn}(\nu_{vu}(t)) = -1\} = \{u \in \mathcal{N}(v) \mid \mathrm{sgn}(\nu_{uv}(t)) = 1\}. \tag{14}$$

While incompressibility of a single flow field $\nu_{uv}(t, \cdot)$ implies the independence of the spatial variable $z \in \Omega_{uv}$, the incompressibility of the overall flow field $\boldsymbol{\nu} = (\nu_e)_{e \in E}$ translates to the fact that the sum of flow rates $q_{uv}(t) = a_{uv}\nu_{uv}(t)$ into a node $v \in V$ is equal to the sum of the flow rates $q_{vu}(t) = a_{vu}\nu_{vu}(t)$ out of $v$:

$$\sum_{u \in \mathcal{N}_-(v,t)} q_{uv}(t) = \sum_{u \in \mathcal{N}_+(v,t)} q_{vu}(t).$$

**The Challenges of Advection-dominated Dynamics on Metric Graphs** Advection-dominated dynamical systems on metric graphs, entail the challenge of complex, antisymmetric, and spatio-temporal dependency structures. We want visualize this using the simple example of an advection metric path graph, that is, a sequence of nodes $(v_0, v_1, ..., v_n)$ with edges $e_{j(j+1)}$ in-between two consecutive nodes $v_j$ and $v_{j+1}$. We assume that the corresponding edge spaces $\Omega_{j(j+1)}$ are of equal size $l_{j(j+1)} = 1m$ and have a constant flow field of $\nu_{j(j+1)} = 2m/s$ for all $e_{j(j+1)} \in E$. If we access information at each of the nodes $v_j$ at discrete time points $t_0, ..., t_i, t_{i+1}, ...$ with temporal discretization $dt = t_{i+1} - t_i = 60s$, information travels $dt \cdot \nu_{j(j+1)} = 60s \cdot 2m/s = 120m$ per time step $dt$, which corresponds to $\frac{dt \cdot \nu_{j(j+1)}}{l_{j(j+1)}} = 120$ hops in the path graph. By this, the signal $c_v(t_i)$ at a node $v$ at time $t_i$ depends on the signal $c_u(t_j)$ at time $t_j$ and a node $u$ which lies 120 hops upstream of $v$. If such path does not exist, the path will be shorter and will originate at boundary nodes. The above example provides a sketch how the spatial dependency structure depends on the lengths, the flow field and the temporal discretization.

For more complex structured graphs and more complex configurations of PDE-parameters, the spatial dependency structure will be a tree that extends from the target (root) node $v$ at time $t$ to upstream nodes (leafs) as the node $u$ in the example above (also see Appendix C.2). The dependency tree will extend further in directions with high flow rate and less in direction where the flow rate is lower. This results in two major challenges for standard MPNNs: Firstly, the extent of the receptive field can vary drastically across the space of the metric graph and over time due to varying PDE-parameters, making it impossible to find a single suitable number of message passing layers. Secondly, the receptive field can be highly asymmetric. These two properties require highly selective and anisotropic message passing that depends on the structure of the metric graph, flow field and the temporal discretization.

While there already exist work that discusses graphs and GNNs specifically taylored to advection problems (Chapman, 2015; Eliasof et al., 2024), the underlying graph structure corresponds to a discretization of the Euclidean space; a paradigm different from the one on metric graphs (cf. Section 2). We highlight this difference in Figure 10.

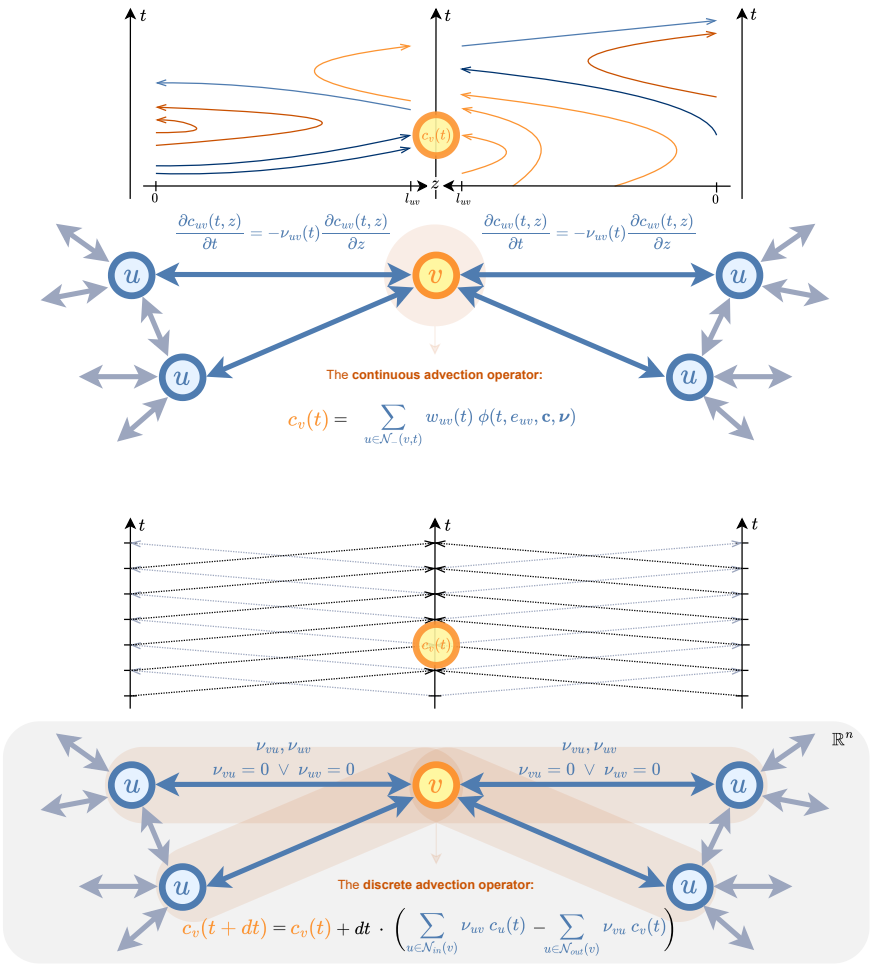



Figure 10: Advection on a metric graph (top) vs. on a non-metric graph (bottom): A metric graph is a system of one-dimensional subspaces $\Omega_{uv} = (0, l_{uv})$, corresponding to edges $e_{uv} \in E$, which are coupled at nodes $V$. Each edge $e_{uv}$ is associated with a one-dimensional PDE that defines the dynamics of the signal $c_{uv}(t, z)$ over time $t \in \mathbb{R}$ and along the edge space $z \in \Omega_{uv} = (0, l_{uv})$, parametrized by a known function $\nu_{uv}$. In case of advection, the known functions $\boldsymbol{\nu} = (\nu_{uv})_{e_{uv} \in E}$ correspond to the time-dependent flow field of the metric graph affecting the spatio-temporal evolution of the signals $c_{uv}$ (dark blue, light blue, dark orange and light orange lines). The coupling condition that defines the signal $c_v(t)$ at a node $v \in V$ and over time $t \in \mathbb{R}$ can be expressed as a weighted aggregation of features $\phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu}) = c_{uv}(t, l_{uv})$ at the boundary $z = l_{uv}$ (orange-shaded ball around $v$) of the edge spaces $\Omega_{uv}$ associated with incoming edges $e_{uv}$ (cf. Eq. (6)). Leveraging the edge dynamics, their value can be traced back to either features $c_u(t - \delta t_{uv})$ of neighboring nodes $u \in \mathcal{N}_-(v, t)$ (dark blue lines) or features $c_v(t - \delta t_{uv})$ of the node itself (light orange lines) and at earlier, over the time and neighboring nodes varying times $t - \delta t_{uv}$ (cf. Eq. (9)).
In contrast, a (conventional, non-metric) graph can be considered as a discretization of the spatial Euclidean space $\Omega = \mathbb{R}^n$ (gray area). Here, edges $e_{uv} \in E$ preserve proximity in between nodes $V$, corresponding to coordinates in the Euclidean domain. The overall space $\Omega = \mathbb{R}^n$ is associated with a multi-dimensional PDE that defines the dynamics of the signal $c(t, v) := c_v(t)$ over time $t \in \mathbb{R}$ and space $v \in V \subset \Omega$, parametrized by a function $\boldsymbol{\nu}$. In case of advection, the discretized advection-operator of the underlying multi-dimensional and continuous advection-PDE $\frac{\partial c_v(t)}{\partial t} = -\boldsymbol{\nu}(t) \cdot \nabla c_v(t, \mathbf{z})$ defines a node update $c_v(t + dt)$ that aggregates the net inflow from neighboring nodes $u \in \mathcal{N}_{\text{in}}(v)$ minus the outflow to neighboring nodes $u \in \mathcal{N}_{\text{out}}(v)$ and at a fixed time $t \in \mathbb{R}$ (black and gray lines and orange-shaded area aorund $v$). In practice, the discretized version $\boldsymbol{\nu} = (\nu_{uv})_{e_{uv} \in E}$ of the continuous flow field is a learned parameter of a GNN.

# B Details on the Method

**Message Passing** has become a common principle in GNNs (Gilmer et al., 2017). It describes how local information at the nodes of a graph is passed to neighboring nodes. Message passing as defined in Gilmer et al. (2017) consists of three consecutive steps: Message generation, message aggregation, and node update. Message generation describes how a feature vector at one node is transformed as it is sent to a neighbor. Message aggregation and node update describe how multiple messages arriving at a node are accumulated into a single feature vector and update the features at the target node. A general formulation of this method is given by

$$\mathbf{x}_v(t+1) = \theta\left(\sum_{u\in\mathcal{N}(v)} \phi\big(\mathbf{x}_u(t), \mathbf{x}_v(t), \mathbf{q}_{uv}(t)\big),\ \mathbf{x}_v(t)\right), \tag{15}$$

where $\phi$ is the message generation function, $\sum$ is a permutation-invariant aggregation function (e.g. sum or mean), $\theta$ is a function that updates the features $\mathbf{x}_v$ of node $v \in V$, and $\mathbf{q}_{uv}$ are optional edge features of the edge $e_{uv} \in E$ from node $u \in \mathcal{N}(v)$ to node $v \in V$. Moreover, $t$ denotes the message-passing iteration step.

## B.1 Final Algorithm: MeGA-MP

**Interpolation** Different interpolation schemes can be applied for time warping, depending on the use case. A version of linear interpolation of a value $c_u(t - \delta t_{uv})$ can be modeled as

$$c_u(t - \delta t_{uv}) \approx\ a(t_j) \cdot c_u(t_j) + a(t_{j+1}) \cdot c_u(t_{j+1}) \tag{16}$$

with $a(t_{j+1}) := \frac{t-\delta t_{uv}-t_j}{t_{j+1}-t_j}$ and $a(t_j) := 1 - \frac{t-\delta t_{uv}-t_j}{t_{j+1}-t_j}$ for the $j \in I$ such that $t - \delta t_{uv} \in [t_j, t_{j+1})$ holds.

**Model Iterations** Based on Equation (16), we formally define $\varphi(t, e_{uv}, \hat{\mathbf{c}}^{(k)}, \boldsymbol{\nu})$ from Equation (11) as

$$\varphi(t, e_{uv}, \hat{\mathbf{c}}^{(k)}, \boldsymbol{\nu}) := \begin{cases} a(t_j) \cdot \hat{c}_u^{(k)}(t_j) + a(t_{j+1}) \cdot \hat{c}_u^{(k)}(t_{j+1}) & \text{if } \not{z}(\delta t_{uv}) \in \{-l_{uv}, l_{uv}\} \\ a(t_j) \cdot \hat{c}_v^{(k)}(t_j) + a(t_{j+1}) \cdot \hat{c}_v^{(k)}(t_{j+1}) & \text{if } \not{z}(\delta t_{uv}) = 0 \end{cases} \tag{17}$$

for the $j \in I$ for which $t - \delta t_{uv} \in [t_j, t_{j+1})$ holds. In practice, we precompute the transport times $\delta t_{uv}$ and implement $\varphi$ as an interpolation operator $\mathcal{I}[c_w^{(k)}](t - \delta t_{uv})$ at the correct node $w \in \{u, v\}$ that does not depend on the flow field $\boldsymbol{\nu}$ anymore (cf. Algorithm 2).

Further details on the implementation can be found in algorithms 1 and 2.

**Algorithm 1** Preprocessing: Converting flows $\nu_{uv}(t)$ to transport times $\delta t_{uv}(t)$ (Definition C.6). This algorithm is implemented in a parallelized/vectorized form for efficiency.

**Inputs:** Flow velocities $\nu_{uv} \in \mathbb{R}^n$ for edge $e_{uv} \in E$ for $n$ time steps ($n = n_{\text{hi}} + n_{\text{fu}}$); Edge length $l_{uv} \in \mathbb{R}$ for edge $e_{uv} \in E$.
$C \leftarrow \text{cumsum}(\nu_{uv}) \in \mathbb{R}^{n+1}$ ▷ cumulative distance traveled over time, with $C_0 = 0$
$\Delta T \in \mathbb{R}^n$ init. empty ▷ array to store transport times
$M^{sl} \in \{0,1\}^n$ init. zero ▷ array to store if self-loop occurs at time $t$
▷ Compute (inverse) pass-through transport times
**for** $t = 0$ **to** $n-1$ **do** ▷ Trace characteristic curve starting at time $t$
  **for** $x = t$ **down to** $0$ **do** ▷ Trace backward along time steps
    $d_{x,t} \leftarrow C_t - C_x$ ▷ distance traveled between $x$ and $t$
    **if** $d_{x,t} \geq l_{uv}$ **then** ▷ true for pass-throughs
      interpolate $\hat{x}$ between $C_x$ and $C_{x+1}$ if needed
      $\Delta T_t \leftarrow t - \hat{x}$ **; break**
    **else if** $d_{x,t} \leq -l_{uv}$ **then** ▷ true for inverse pass-throughs
      interpolate $\hat{x}$ between $C_x$ and $C_{x+1}$ if needed
      $\Delta T_t \leftarrow \hat{x} - t$ **; break**
    **end if**
  **end for**
**end for** ▷ Compute self-loop transport times
**for** $t = 1$ **to** $n-1$ **do** ▷ Trace characteristic curve starting at time $t$
  **for** $x = t$ **down to** $1$ **do**
    $d_1 \leftarrow C_{x-1} - C_t$; $d_2 \leftarrow C_x - C_t$ ▷ distance differences between consecutive steps
    **if** $\text{sign}(d_1) \neq \text{sign}(d_2)$ **then** ▷ zero-crossing indicates self-loop
      interpolate zero crossing: $\hat{x} \leftarrow x - d_2/(d_2 - d_1)$
      $\Delta T_t \leftarrow \text{sign}(d_2) \cdot (t - \hat{x})$
      $M^{sl}_t \leftarrow 1$ **; break**
    **end if**
  **end for**
**end for**
**Output:** $\Delta T \in \mathbb{R}^n$; $M^{sl} \in \{0,1\}^n$ ▷ $\Delta T_t > 0 \Leftrightarrow$ transport in edge dir., $\Delta T_t < 0 \Leftrightarrow$ transport opposing edge dir.

**Algorithm 2** MeGA-MP: Metric Graph Advection Message Passing

**Inputs:** Graph structure $\mathcal{G} = (V, E)$; Initial node states / known history $x \in \mathbb{R}^{|V| \times k_{hi}}$; Transport times $\delta t_{uv}(t)$ for each edge $e_{uv} \in E$ and for each time step $t_i \in T$, where $|T| = n = n_{\text{hi}} + n_{\text{fu}}$.
$x^{(0)} \leftarrow \text{pad}(x, n_{\text{fu}})$ ▷ pad initial condition by prediction length $n_{\text{fu}}$
Apply Dirichlet boundary conditions to $x^{(0)}$ at boundary nodes
$x_{\text{out}} \leftarrow x^{(0)}$ ▷ initialize output accumulator
**for** $i = 1$ **to** $\kappa$ **do** ▷ Iterate until convergence, c.f. Theorem 3.6
  **for** each node $v \in V$ and time $t$ **do** ▷ Message-Passing
    $x^{(i)}_{v,t} \leftarrow \sum_{u \in \mathcal{N}(v)} w_{uv}(t)(\mathcal{I}[x^{(i-1)}_u](t - \delta t_{uv}(t))\mathbb{1}_{M^{sl}_{uv}(t)=0}$
    $+ \quad \mathcal{I}[x^{(i-1)}_v](t - \delta t_{uv}(t))\mathbb{1}_{M^{sl}_{uv}(t)=1})$ ▷ cf. Eq. 6 and Sec. 3.2 and B.1
  **end for**
  $x^{(i)}_{v,t} \leftarrow 0 \quad \forall v \in V_b, \forall t \in T$ ▷ Set to zero at boundary nodes.
  $x_{\text{out}} \leftarrow x_{\text{out}} + x^{(i)}$
**end for**
**if** boundary values exist **then**
  Set $x_{\text{out}}$ at boundary nodes
**end if**
**Output:** $x_{\text{out}}$ ▷ final node states after advection message passing

# C Details on Theoretical Guarantees

## C.1 Physical Derivation of MeGA-MP

While in Section 3.1, we provide an intuition on how MeGA-MP incorporates the physical priors from Section 2, in this section, we present a rigorous derivation of our metric graph advection message passing algorithm based on these priors. More precisely, we prove that Equation (10) is equal to the mixing at nodes (Eq. (4)) given the advection dynamics (Eq. (2)) with inflow continuity condition (Eq. (5)) and suitable smoothness assumptions (Assumption (3.4)). We repeat these priors for convenience while giving an overview of the rest of this section.

In Section C.1.1, we derive the message-generation function $\phi$ as defined in Equation (9) based on the *advection edge dynamics*. For each $e_{uv} \in E$, this refers to the change of the concentration $c_{uv} : \mathbb{R} \times \Omega_{uv} \to \mathbb{R}_{\geq 0}$ over time $\mathbb{R}$ and along the edge space $\Omega_{uv} = (0, l_{uv})$, determined by the incompressible *flow field* $\nu_{uv} : \mathbb{R} \to \mathbb{R}$ over time $\mathbb{R}$. Formally, this is defined by the PDE

$$\frac{\partial c_{uv}(t,z)}{\partial t} = -\nu_{uv}(t)\, \frac{\partial c_{uv}(t,z)}{\partial z} \qquad \text{(Equation (2))}.$$

In Section C.1.2, we derive the message aggregation and node update scheme $c_v(t)$ as defined in Equation (10) based on the *coupling conditions* given by the *mixing at nodes* and the *inflow continuity condition*. For each $v \in V$, the former refers to the concentration $c_v : \mathbb{R} \to \mathbb{R}_{\geq 0}$ over time $\mathbb{R}$, determined by a weighted aggregation of the concentrations $c_{uv}(\cdot, l_{uv})$ carried by the *flow rates* $q_{uv} = \alpha_{uv}\nu_{uv} : \mathbb{R} \to \mathbb{R}$ from (temporal) inflow neighbors $u \in \mathcal{N}_-(v, \cdot)$ over time $\mathbb{R}$. Formally, this is defined as

$$c_v(t) = \frac{\sum_{u \in \mathcal{N}_-(v,t)} q_{uv}(t) \cdot c_{uv}(t, l_{uv})}{\sum_{u \in \mathcal{N}_-(v,t)} q_{uv}(t)} \qquad \text{(Equation (4))}.$$

For each edge $e_{uv} \in E$, the latter refers to the concentration $c_{uv} : \mathbb{R} \times \Omega_{uv} \to \mathbb{R}_{\geq 0}$ over time $\mathbb{R}$ and at the temporal inflow boundary $z = 0$ of the edge space $\Omega_{uv} = (0, l_{uv})$, determined by the concentration $c_u : \mathbb{R} \to \mathbb{R}$ of the temporal inflow neighbor $u \in \mathcal{N}_-(v, \cdot)$ over time $\mathbb{R}$. Formally, this is defined by

$$c_{uv}(t, 0) = c_u(t) \text{ for all } u \in \mathcal{N}_-(v, t) \qquad \text{(Equation (5))}.$$

The mixing at nodes and the inflow continuity condition formalize the following intuition: On the one hand, if there is a (physical) flow from $u \in \mathcal{N}_-(v,t)$ to $v \in V$ entering the edge $e_{uv} \in E$ at time $t \in \mathbb{R}$, the concentration $c_{uv}(t, 0)$ at the beginning $z = 0$ of that edge $e_{uv}$ is equal to the concentration $c_u(t)$ at the node $u$ out of which the flow flows *into that edge* (*local inflow boundary condition*). On the other hand, if there is a (physical) flow from $u \in \mathcal{N}_-(v,t)$ to $v \in V$ leaving the edge $e_{uv} \in E$ at time $t \in \mathbb{R}$, the concentration $c_{uv}(t, l_{uv})$ at the end $z = l_{uv}$ of that edge $e_{uv}$ contributes to the weighted aggregation at the node $v$ into which the flow flows *out of that edge* (*local outflow boundary condition*).

While the edge dynamics (Eq. (1)) are defined on the open edge spaces $\Omega_{uv} = (0, l_{uv})$, the coupling conditions (Eq. (4) and Eq. (5)) rely on the values $c_{uv}(\cdot, 0)$ and $c_{uv}(\cdot, l_{uv})$ on the boundary $\partial\Omega_{uv} = \{0, l_{uv}\}$ of $\Omega_{uv}$. Therefore, suitable smoothness assumptions on $c_{uv}$ and $\nu_{uv}$ are required to be able to transfer information in-between the edge spaces and its coupling boundaries.[3]

**Assumption** (Assumption 3.4)**.** For each $e_{uv} \in E$, let

- $c_{uv} : \mathbb{R} \times (0, l_{uv}) \to \mathbb{R}_{\geq 0}$ be differentiable,
- $c_{uv} : \mathbb{R} \times [0, l_{uv}] \to \mathbb{R}_{\geq 0}$ be continuous and
- $\nu_{uv} : \mathbb{R} \to \mathbb{R}$ be continuous.

[3]Note that strictly speaking, we also need some part of the assumptions simply for Equation (2) or more generally, Equation (1), to be well-defined.

*Remark* C.1 (Symmetry of signals). Note that next to the physical priors introduced above, simply because of the fact that we model a metric graph as symmetric directed graph $\mathcal{G}$ (cf. Section 2),

$$c_{vu}(t,z) = c_{uv}(t, l_{vu} - z) \tag{18}$$

holds for all $v \in V$, $u \in \mathcal{N}(v)$, $t \in \mathbb{R}$ and $z \in \overline{\Omega}_{vu} = [0, l_{vu}]$. That is, given that we consider a symmetric directed graph $\mathcal{G} = (V, E)$ and the contrasting edges $e_{vu}, e_{uv} \in E$ of length $l_{vu} = l_{uv}$, we expect that the signal $c_{vu}(t,z)$ along the edge $e_{vu}$ at time $t$ and position $z \in [0, l_{vu}]$ is equal to the signal along the edge $e_{uv}$ at time $t$ and position $l_{vu} - z = l_{uv} - z \in [0, l_{uv}] = \overline{\Omega}_{uv}$.

### C.1.1 Message Generation: Advective Transport Along Edges

Throughout the rest of this section, we assume that for each $e_{uv} \in E$ (that is, for each $v \in V$ and $u \in \mathcal{N}(v)$), Equation (2) with coupling conditions as defined by Equation (4) and (5) and smothness assumptions as defined by Assumption 3.4 holds. According to Equation (6), the message-generation function $\phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu}) = c_{uv}(t, l_{uv})$ relies on the signal $c_{uv}(\cdot, l_{uv})$ over time $\mathbb{R}$ and at the temporal *outflow* boundary $z = l_{uv}$ of the edge space $\Omega_{uv} = (0, l_{uv})$. Since we are only aware of the signal $c_{uv}(\cdot, 0)$ over time $\mathbb{R}$ and at the temporal *inflow* boundary $z = 0$ of each edge space $\Omega_{uv} = (0, l_{uv})$ through the inflow boundary condition (Eq. (5)), computing $\phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu})$ would normally require to explicitly solve the edge dynamics along the whole edge space $\overline{\Omega}_{uv} = [0, l_{uv}]$. This is typically done by computationally expensive numerical solvers. However, our message-generation function $\phi$ builds upon the MoC to avoid the explicit computation of concentrations $c_{uv}(\cdot, z)$ in between the boundaries, i.e., for all $z \in (0, l_{uv})$. More specifically, we can derive *characteristic curves* along which the function $c_{uv}$ remains constant over time $\mathbb{R}$ and space $\Omega_{uv}$.

**Lemma C.2** (Constant concentration along characteristic curves (full version))**.** *Let $e_{uv} \in E$ hold. If the function $c_{uv}$ obeys Equation* (2)*, for each $t_0 \in \mathbb{R}$ and $z_0 \in \mathbb{R}$, the curve*

$$\begin{aligned} \gamma_{t_0,z_0} : \; T_{t_0,z_0} &\longrightarrow \mathbb{R}_{\geq 0} \\ t &\longmapsto c_{uv}\left(t, z_0 + \textstyle\int_{t_0}^{t} \nu_{uv}(s)\; ds\right) \end{aligned}$$

*is constant on*

$$T_{t_0,z_0} = \Big\{ t \in \mathbb{R} \mid 0 < z_0 + \int_{t_0}^{t'} \nu_{uv}(s)\; ds < l_{uv} \;\forall t' \in \big[\min\{t_0, t\}, \max\{t_0, t\}\big] \Big\}.$$

Intuitively, in Lemma 3.2, we fix an arbitrary time $t_0 \in \mathbb{R}$ and a position $z_0 \in \mathbb{R}$ and follow any particle that is at this position $z_0$ at this time $t_0$ along the path $z(t) := z_0 + \int_{t_0}^{t} \nu_{uv}(s)\; ds$ it travels during the time $t - t_0$ (or $t_0 - t$) for some time $t \in T_{t_0,z_0}$ with possibly varying flow velocities $\nu_{uv}(s)$ for all $s \in [\min\{t_0, t\}, \max\{t_0, t\}]$. If we assign a concentration $c_{uv}(t_0, z_0)$ to this particle, the concentration should stay constant along the path this particle travels. The choice of $t \in T_{t_0,z_0}$ guarantees that the before-mentioned path remains within the edge $e_{uv}$, or in other words, in the open sub-domain $\mathbb{R} \times \Omega_{uv}$ of the function $c_{uv}$ on which Equation (2) is defined on, as we will investigate in detail in the next Remark C.3.

*Remark* C.3 (On $T_{t_0,z_0}$). Since for $e_{uv} \in E$, the velocity $\nu_{uv} : \mathbb{R} \to \mathbb{R}$ can change its sign, for each $t_0 \in \mathbb{R}$ and $z_0 \in \mathbb{R}$, the function

$$z : \mathbb{R} \longrightarrow \mathbb{R},\; t \longmapsto z(t) := z_0 + \int_{t_0}^{t} \nu_{uv}(s)\; ds$$

is differentiable (and thus, continuous), but *not* monotone, meaning that even if $z(t_0) = z_0 \in \Omega_{uv}$ and $z(t) \in \Omega_{uv}$ holds for a $t \in \mathbb{R}$, this does *not* automatically mean that $z(t') \in \Omega_{uv}$ holds for all $t' \in [\min\{t_0, t\}, \max\{t_0, t\}]$, too. However, since Equation (2) only holds on the open sub-domain $\mathbb{R} \times \Omega_{uv}$ of the function $c_{uv} : \mathbb{R} \times [0, l_{uv}] \to \mathbb{R}_{\geq 0}$, any path from $z(t_0) = z_0$ to $z(t) = z_0 + \int_{t_0}^{t} \nu_{uv}(s)\; ds$ parametrized by time needs to be fully included in the space interval $\Omega_{uv} = (0, l_{uv})$. In other words, $t \in T_{t_0,z_0}$ needs to hold.

Since the set $T_{t_0,z_0}$ can also be empty, we are interested in when this is not the case and what properties it satisfies in this case. The following Lemma C.4 answers these questions.

**Lemma C.4** (On $T_{t_0,z_0}$)**.** *Let $e_{uv} \in E$, $t_0 \in \mathbb{R}$, $z_0 \in \mathbb{R}$ hold and let $T_{t_0,z_0}$ be defined as in Lemma 3.2.*

1. *If $z_0 \notin \Omega_{uv}$ holds, then $T_{t_0,z_0} = \emptyset$ holds.*
2. *If $z_0 \in \Omega_{uv}$ holds, then $T_{t_0,z_0} \neq \emptyset$ holds. Even more, $t_0 \in T_{t_0,z_0}$ holds.*
3. *If $z_0 \in \Omega_{uv}$ holds, there exist $t_l \in [-\infty, t_0)$ and $t_r \in (t_0, \infty]$ such that*
   *(a) if $t_l \neq -\infty$ holds, $z_0 + \int_{t_0}^{t_l} \nu_{uv}(s)\ ds \in \{0, l_{uv}\}$ holds,*
   *(b) if $t_r \neq +\infty$ holds, $z_0 + \int_{t_0}^{t_r} \nu_{uv}(s)\ ds \in \{0, l_{uv}\}$ holds and*
   *(c) $T_{t_0,z_0} = (t_l, t_r) = (t_0 - (t_0 - t_l), t_0 + (t_r - t_0))$ holds.*

   *Equivalently, there exist $\delta t_l, \delta t_r \in (0, \infty]$ such that $T_{t_0,z_0} = (t_0 - \delta t_l, t_0 + \delta t_r)$ holds.*

Since according to the previous Lemmas 3.2 and C.4, for any $t_0 \in \mathbb{R}$ and $z_0 \in \Omega_{uv}$, the concentration $c_{uv}$ is constant along characteristic curves $\gamma_{t_0,z_0}$ on an non-empty time interval $T_{t_0,z_0} = (t_\mathrm{l}, t_\mathrm{r})$, it suffices to know the value of $c_{uv}$ at one position in this time interval to have knowledge of the concentration over all times in the interval:

**Theorem C.5** (Advective transport along edges)**.** *Let $e_{uv} \in E$ hold. If the function $c_{uv}$ obeys Equation* (2)*, for each $t_0 \in \mathbb{R}$ and $z_0 \in \Omega_{uv}$,*

$$c_{uv}\left(t, z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds\right) = c_{uv}\left(t, z_0 + \overline{\nu}_{uv}(t_0, t)\ (t - t_0)\right) = c_{uv}(t_0, z_0)$$

*holds for all $t \in T_{t_0,z_0}$ with $T_{t_0,z_0}$ as defined in Lemma 3.2 and for the mean velocity*

$$\overline{\nu}_{uv}(t_0, t) := \begin{cases} \fint_{t_0}^{t} \nu_{uv}(s)\ ds & \text{if } t \neq t_0 \\ 0 & \text{if } t = t_0 \end{cases} = \begin{cases} \frac{1}{t - t_0} \int_{t_0}^{t} \nu_{uv}(s)\ ds & \text{if } t \neq t_0 \\ 0 & \text{if } t = t_0 \end{cases}.$$

According to Lemma C.4.3, for any $t_0 \in \mathbb{R}$ and $z_0 \in \Omega_{uv}$, the set $T_{t_0,z_0}$ usually corresponds to a time interval $T_{t_0,z_0} = (t_\mathrm{l}, t_\mathrm{r})$ such that on its boundary $\partial(t_\mathrm{l}, t_\mathrm{r}) = \{t_\mathrm{l}, t_\mathrm{r}\}$, the information reaches exactly one of the ends of the edge $e_{uv}$, i.e., the node $v \in V$ or the node $u \in \mathcal{N}(v)$. In view of the coupling conditions (Eq. (4)), this motivates to leverage the advective transport along edges, i.e., Theorem C.5, and combine it with the inflow boundary conditions defined through Equation (5), in order to derive a representation of the advection coupling conditions (Eq. (4)) that does not depend on the concentrations $c_{uv}(t, l_{uv})$ at the end $z = l_{uv}$ of the edges $e_{uv}$ for each $u \in \mathcal{N}_-(v, t)$, but on the concentrations $c_u(t - \delta t_{uv})$ and $c_v(t - \delta t_{uv})$ at suitable times $t - \delta t_{uv} \in \mathbb{R}$. This, in turn, will serve as the basis for our message-passing algorithm (cf. Eq. (6) and (9)).

Intuitively, the time shift $\delta t_{uv} \in \mathbb{R}$ will correspond to the time that any information requires to travel along the edge $e_{uv}$ with the flow $\nu_{uv} : \mathbb{R} \to \mathbb{R}$ from one of the nodes $v \in V$ or $u \in \mathcal{N}(v)$ until it reaches the node $v \in V$ again. From the sender perspective, two different scenarios can occur: Either, the information leaves the node $u \in \mathcal{N}(v)$ at time $t - \delta t_{uv} \in \mathbb{R}$ and enters the node $v \in V$ at time $t \in \mathbb{R}$. We will call this scenario a *pass-through*. Or, the information leaves the node $v \in V$ at time $t - \delta t_{uv} \in \mathbb{R}$ and starts traveling to the node $u \in \mathcal{N}(v)$, but before it reaches that node, the flow $\nu_{vu} = -\nu_{uv} : \mathbb{R} \to \mathbb{R}$ changes its sign and the information enters the node $v \in V$ at time $t \in \mathbb{R}$ again. We will call this scenario a *self-loop*. Of course, the sign of the flow can also change multiple times before the information enters the node $v \in V$. As a first step, we now formalize the aforementioned pass-through and self-loop times formally and collect some helpful properties to get a deeper understanding of such. In practice, also corresponding inverse phenomena will appear. In summary, we call these times *transport times*, a name which we will motivate later.

**Definition C.6** (Transport times)**.** Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. If there exists a $\delta t_{uv} \in \mathbb{R}$, such that

$$\int_{t - \delta t_{uv}}^{t} \nu_{uv}(s)\ ds = l_{uv} \text{ and} \tag{19}$$

$$\int_{t-\delta t_{uv}}^{t'} \nu_{uv}(s)\ ds \in (0, l_{uv}) \tag{20}$$

hold for all $t' \in (\min\{t-\delta t_{uv}, t\}, \max\{t-\delta t_{uv}, t\})$, we call $\delta t_{uv}$ the *pass-through time w.r.t. the edge* $e_{uv} \in E$ *and time* $t \in \mathbb{R}$. If instead, there exists a $\delta t_{uv} \in \mathbb{R}$, such that

$$\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds = -l_{uv} \text{ and} \tag{21}$$

$$\int_{t-\delta t_{uv}}^{t'} \nu_{uv}(s)\ ds \in (-l_{uv}, 0) \tag{22}$$

hold for all $t' \in (\min\{t-\delta t_{uv}, t\}, \max\{t-\delta t_{uv}, t\})$, we call $\delta t_{uv}$ the *inverse pass-through time w.r.t. the edge* $e_{uv} \in E$ *and time* $t \in \mathbb{R}$. If instead, there exists a $\delta t_{uv} \in \mathbb{R}_{\neq 0}$, such that

$$\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds = 0 \text{ and} \tag{23}$$

$$\int_{t-\delta t_{uv}}^{t'} \nu_{uv}(s)\ ds \in (0, l_{uv}), \tag{24}$$

hold for all $t' \in (\min\{t-\delta t_{uv}, t\}, \max\{t-\delta t_{uv}, t\})$, we call $\delta t_{uv}$ the *self-loop time w.r.t. the edge* $e_{uv} \in E$ *and time* $t \in \mathbb{R}$. If instead, there exists a $\delta t_{uv} \in \mathbb{R}_{\neq 0}$, such that

$$\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds = 0 \text{ and} \tag{25}$$

$$\int_{t-\delta t_{uv}}^{t'} \nu_{uv}(s)\ ds \in (-l_{uv}, 0), \tag{26}$$

hold for all $t' \in (\min\{t-\delta t_{uv}, t\}, \max\{t-\delta t_{uv}, t\})$, we call $\delta t_{uv}$ the *inverse self-loop time w.r.t. the edge* $e_{uv} \in E$ *and time* $t \in \mathbb{R}$.

Usually, $e_{uv} \in E$ and $t \in \mathbb{R}$ are clear from the context and we just say *pass-through time*, *inverse pass-through time*, *self-loop time* or *inverse self-loop time*.

The following lemmas help to get a deeper understanding of the different kinds of transport times.

**Lemma C.7** (Pass-through time)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. If there exists a pass-through time $\delta t_{uv} \in \mathbb{R}$ w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$, the following properties hold:*

1. *$\delta t_{uv} \in \mathbb{R}_{\neq 0}$ and thus, $t - \delta t_{uv} \neq t$ holds.*
2. *There exists no other pass-through time of the same sign (i.e., there exists either exactly one or no positive pass-through time, and there exists exactly one or no negative pass-through time).*
3. *The mean velocity*

$$\overline{\nu}_{uv}(t) := \overline{\nu}_{uv}(t-\delta t_{uv}, t) = \fint_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds = \frac{1}{\delta t_{uv}} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds$$

*during the time interval $[\min\{t-\delta t_{uv}, t\}, \max\{t-\delta t_{uv}, t\}]$ as defined in Theorem C.5 is equal to $\overline{\nu}_{uv}(t) = \frac{l_{uv}}{\delta t_{uv}}$ and thus satisfies*

$$\overline{\nu}_{uv}(t) > 0 \Longleftrightarrow \delta t_{uv} > 0 \text{ and}$$
$$\overline{\nu}_{uv}(t) < 0 \Longleftrightarrow \delta t_{uv} < 0.$$

4. *The flows* $q_{uv}(t-\delta t_{uv})$ *and* $q_{uv}(t)$ *linked to the velocity* $\nu_{uv}(\cdot) = \frac{q_{vu(\cdot)}}{\alpha_{vu}}$ *satisfy*

$$\begin{cases} q_{uv}(t-\delta t_{uv}), q_{uv}(t) \geq 0 & \text{if } \delta t_{uv} > 0 \\ q_{uv}(t-\delta t_{uv}), q_{uv}(t) \leq 0 & \text{if } \delta t_{uv} < 0 \end{cases}.$$

According to Lemma C.7, if a pass-through time $\delta t_{uv} > 0$ is positive, so does $\overline{\nu}_{uv}(t) > 0$ hold, and there is an average flow from the node $u \in \mathcal{N}(v)$ to the node $v \in V$ during the time $[t-\delta t_{uv}, t]$.[4] If in contrast, a pass-through time $\delta t_{uv} < 0$ is negative, so does $\overline{\nu}_{uv}(t) < 0$ hold, and there is an average flow from the node $v \in V$ to the node $u \in \mathcal{N}(v)$ during the time $[t, t-\delta t_{uv}]$.[5] In both cases, the time $\delta t_{uv} = \frac{l_{uv}}{\overline{\nu}_{uv}(t)}$ is exactly the time that any particle transported by the average flow $\overline{q}_{uv}(t) = \alpha_{uv}\ \overline{\nu}_{uv}(t)$ needs to travel along the edge $e_{uv}$ of length $l_{uv}$. Therefore, we call $\delta t_{uv}$ the *pass-through time*. The cases are intuitively connected by the equality[6]

$$\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds = \int_{t}^{t-\delta t_{uv}} -\nu_{uv}(s)\ ds = \int_{t}^{t-\delta t_{uv}} \nu_{vu}(s)\ ds,$$

where in dependence of the sign of the pass-through time $\delta t_{uv} \in \mathbb{R}_{\neq 0}$, we use the oriented of the two integrals on the left- or right-hand side for interpretation. We can make similar observations for self-loop times.

**Lemma C.8** (Self-loop time)**.** *Let* $e_{uv} \in E$ *and* $t \in \mathbb{R}$ *hold. If there exists a self-loop time* $\delta t_{uv} \in \mathbb{R}_{\neq 0}$ *w.r.t.* $e_{uv} \in E$ *and* $t \in \mathbb{R}$*, the following properties hold:*

1. *There exists no other self-loop time of the same sign (i.e., there exists either exactly one or no positive self-loop time, and there exists exactly one or no negative self-loop time).*
2. *The mean velocity*

$$\overline{\nu}_{uv}(t) := \overline{\nu}_{uv}(t-\delta t_{uv}, t) = \fint_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds = \frac{1}{\delta t_{uv}} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds$$

*during the time interval* $[\min\{t-\delta t_{uv}, t\}, \max\{t-\delta t_{uv}, t\}]$ *as defined in Theorem C.5 satisfies* $\overline{\nu}_{uv}(t) = 0$.

3. *The flows* $q_{uv}(t-\delta t_{uv})$ *and* $q_{uv}(t)$ *linked to the velocity* $\nu_{uv}(\cdot) = \frac{q_{vu(\cdot)}}{\alpha_{vu}}$ *satisfy*

$$\begin{cases} q_{uv}(t-\delta t_{uv}) \geq 0,\ q_{uv}(t) \leq 0 & \text{if } \delta t_{uv} > 0 \\ q_{uv}(t-\delta t_{uv}) \leq 0,\ q_{uv}(t) \geq 0 & \text{if } \delta t_{uv} < 0 \end{cases}.$$

According to Lemma C.8, independent on whether a self-loop time $\delta t_{uv} \in \mathbb{R}_{\neq 0}$ is positive or negative, $\overline{\nu}_{uv}(t) = 0$ holds. If a self-loop time $\delta t_{uv} > 0$ is positive, there is an average flow from the node $u \in \mathcal{N}(v)$ to itself during the time $[t-\delta t_{uv}, t]$.[7] If in contrast, a self-loop time $\delta t_{uv} < 0$ is negative, there is an average flow from the node $u \in \mathcal{N}(v)$ to itself during the time $[t, t-\delta t_{uv}]$.[8] In both cases, the time $\delta t_{uv}$ is exactly the time that any particle transported by the average flow $\overline{q}_{uv}(t) = \alpha_{uv}\ \overline{\nu}_{uv}(t) = 0$ needs to travel from $u$ along some part of the edge $e_{uv}$ until the flow changes its sign and the particle travels back to the node $u$ it came from. Therefore, we call $\delta t_{uv}$ the *self-loop time*.

[4] More precisely, since in this case, $q_{uv}(t-\delta t_{uv}), q_{uv}(t) \geq 0$ holds, we know that at time $t-\delta t_{uv} \in \mathbb{R}$ and $t \in \mathbb{R}$, there is a flow from $u \in \mathcal{N}_-(v, t-\delta t_{uv}) \cap \mathcal{N}_-(v,t)$ to $v \in V$. Note that we tacitly ignored the case of zero flows here due to Remark E.1.

[5] More precisely, since in this case, $q_{uv}(t-\delta t_{uv}), q_{uv}(t) \leq 0$ holds, we know that at time $t \in \mathbb{R}$ and $t-\delta t_{uv} \in \mathbb{R}$, there is a flow from $v \in V$ to $u \in \mathcal{N}_+(v,t) \cap \mathcal{N}_+(v, t-\delta t_{uv})$. Note that we tacitly ignored the case of zero flows here due to Remark E.1.

[6] The proof follows similar arguments as given in, e.g., the proof of Lemma C.9.

[7] More precisely, since in this case, $q_{uv}(t-\delta t_{uv}) \geq 0$ and $q_{uv}(t) \leq 0$ holds, we know that at time $t-\delta t_{uv} \in \mathbb{R}$, there is a flow from $u \in \mathcal{N}_-(v, t-\delta t_{uv})$ to $v \in V$, while at time $t \in \mathbb{R}$, there is a flow from $v \in V$ to $u \in \mathcal{N}_+(v,t)$. Note that we tacitly ignored the case of zero flows here due to Remark E.2.

[8] More precisely, since in this case, $q_{uv}(t) \geq 0$ and $q_{uv}(t-\delta t_{uv}) \leq 0$ holds, we know that at time $t \in \mathbb{R}$, there is a flow from $u \in \mathcal{N}_-(v,t)$ to $v \in V$, while at time $t-\delta t_{uv} \in \mathbb{R}$, there is a flow from $v \in V$ to $u \in \mathcal{N}_+(v, t-\delta t_{uv})$. Note that we tacitly ignored the case of zero flows here due to Remark E.2.

As another result that will turn out to be useful in practise, by the symmetry of the underlying graph $\mathcal{G}$ and the conservation of flows (Eq. (3)), an inverse pass-through time induces a pass-through time and vice versa, while an inverse self-loop time induces a self-loop time and vice versa.

**Lemma C.9** (Inverse pass-through time)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. $\delta t_{uv} \in \mathbb{R}$ is an inverse pass-through time w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$ if and only if (iff) $\delta t_{vu} := \delta t_{uv} \in \mathbb{R}$ is a pass-through time w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$*

**Lemma C.10** (Inverse self-loop time)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. $\delta t_{uv} \in \mathbb{R}$ is an inverse self-loop time w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$ iff $\delta t_{vu} := \delta t_{uv} \in \mathbb{R}$ is a self-loop time w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$*

We now collected all relevant properties which help us to prove the main results of this section as motivated above: Given a pass-through time $\delta t_{uv} \in \mathbb{R}$ w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$, we can re-write the concentrations $c_{uv}(t, l_{uv})$ and $c_{vu}(t - \delta t_{uv}, l_{vu})$ at the end $z = l_{uv} = l_{vu}$ of the edges $e_{uv}$ and $e_{vu}$ using the concentrations $c_u : \mathbb{R} \to \mathbb{R}_{\geq 0}$ and $c_v : \mathbb{R} \to \mathbb{R}_{\geq 0}$, respectively:

**Theorem C.11** (Pass-through time)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. If the function $c_{uv}$ obeys Equation (2) and there exists a pass-through time $\delta t_{uv} \in \mathbb{R}_{\neq 0}$ w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$, we obtain*

$$c_{uv}(t, l_{uv}) = c_{uv}(t - \delta t_{uv}, 0) = c_{uv}\left(t - \frac{l_{uv}}{\overline{\nu}_{uv}(t)}, 0\right).$$

*Even more, the concentrations $c_{uv}(\cdot, l_{uv})$ and $c_{vu}(\cdot, l_{vu})$ at the end $z = l_{uv} = l_{vu}$ of the edges $e_{uv} \in E$ and $e_{vu} \in E$ are connected to the concentrations $c_u$ and $c_v$ of the nodes of that edge, respectively, by*

$$\begin{cases} c_{uv}(t, l_{uv}) = c_u(t - \delta t_{uv}) & \text{if } \delta t_{uv} > 0 \\ c_{vu}(t - \delta t_{uv}, l_{vu}) = c_v(t) & \text{if } \delta t_{uv} < 0 \end{cases}.$$

According to Theorem C.11, if a pass-through time $\delta t_{uv} > 0$ is positive, the concentration $c_{uv}(t, l_{uv})$ at time $t \in \mathbb{R}$ and at the end of the edge $e_{uv} \in E$, along which there is an average flow from the node $u \in \mathcal{N}(v)$ to the node $v \in V$ during the time $[t - \delta t_{uv}, t]$, is equal to the concentration $c_u(t - \delta t_{uv})$ at the node $u \in \mathcal{N}(v)$ and $-\delta t_{uv} < 0$ units of time *earlier* than $t$. If in contrast, a pass-through time $\delta t_{uv} < 0$ is negative, the concentration $c_{vu}(t - \delta t_{uv}, l_{vu})$ at time $t - \delta t_{uv} \in \mathbb{R}$ and at the end of the edge $e_{vu} \in E$, along which there is an average flow from the node $v \in V$ to the node $u \in \mathcal{N}(v)$ during the time $[t, t - \delta t_{uv}]$, is equal to the concentration $c_v(t)$ at the node $v \in V$ and $+\delta t_{uv} < 0$ units of time *earlier* than $t - \delta t_{uv}$.

Analogously, given a self-loop time $\delta t_{uv} \in \mathbb{R}$ w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$, we can re-write the concentration $c_{vu}(t, l_{vu})$ at the end $z = l_{vu} = l_{uv}$ of the edge $e_{vu}$ using the concentration $c_u : \mathbb{R} \to \mathbb{R}_{\geq 0}$:

**Theorem C.12** (Self-loop time)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. If the function $c_{uv}$ obeys Equation (2) and there exists a self-loop time $\delta t_{uv} \in \mathbb{R}_{\neq 0}$ w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$, we obtain*

$$c_{uv}(t, 0) = c_{uv}(t - \delta t_{uv}, 0).$$

*Even more, the concentration $c_{vu}(\cdot, l_{vu})$ at the end $z = l_{vu} = l_{uv}$ of the edge $e_{vu} \in E$ is connected to the concentration $c_u$ of the node of that edge by*

$$\begin{cases} c_{vu}(t, l_{vu}) = c_u(t - \delta t_{uv}) & \text{if } \delta t_{uv} > 0 \\ c_{vu}(t - \delta t_{uv}, l_{vu}) = c_u(t) & \text{if } \delta t_{uv} < 0 \end{cases}.$$

According to Theorem C.12, if a self-loop time $\delta t_{uv} > 0$ is positive, the concentration $c_{vu}(t, l_{vu})$ at time $t \in \mathbb{R}$ and at the end of the edge $e_{vu} \in E$, along which there is an average flow from the node $u \in \mathcal{N}(v)$ to itself during the time $[t - \delta t_{uv}, t]$, is equal to the concentration $c_u(t - \delta t_{uv})$ at the node $u \in \mathcal{N}(v)$ and $-\delta t_{uv} < 0$ units of time *earlier* than $t$. If in contrast, a self-loop time $\delta t_{uv} < 0$ is negative, the concentration $c_{vu}(t - \delta t_{uv}, l_{vu})$ at time $t - \delta t_{uv} \in \mathbb{R}$ and at the end of the edge $e_{vu} \in E$, along which there is an average flow from the node $u \in \mathcal{N}(v)$ to itself during the time $[t, t - \delta t_{uv}]$, is equal to the concentration $c_u(t)$ at the

node $u \in \mathcal{N}(v)$ and $+\delta t_{uv} < 0$ units of time *earlier* than $t - \delta t_{uv}$.

In summary, Theorem C.11 and Theorem C.12 illustrate the time shift required to pass information from one node to another. Therefore we call these shifts *transport times*, and the underlying concepts serve as a basis for our advection-message-generation function $\phi$ (cf. Eq. (6)). To be able to properly define it, we need to make sure that for each edge $e_{uv} \in E$ and time $t \in \mathbb{R}$, such a transport time exists and is unique. For the sake of brevity, for such a pair, we define the function

$$\mathscr{z} : \mathbb{R} \longrightarrow \mathbb{R},\ \delta t \longmapsto \mathscr{z}(\delta t) := \int_{t-\delta t}^{t} \nu_{uv}(s)\ ds \quad \text{(Equation (7))},$$

where we omit the dependency of $\mathscr{z}$ on $e_{uv} \in E$, $t \in \mathbb{R}$ and the overall flow field $\boldsymbol{\nu}$ for better readability. Then we obtain the following existence and uniqueness result:

**Theorem C.13** (Existence and uniqueness of positive transport times)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. If $\nu_{uv}(t) \neq 0$ holds and the set*

$$\begin{aligned} A_{uv}(t) :=& \left\{ \delta t \in (0,\infty) \ \middle|\ \int_{t-\delta t}^{t} \nu_{uv}(s)\ ds \in \{-l_{uv}, 0, l_{uv}\} \right\} \\ =& \left\{ \delta t \in (0,\infty) \ \middle|\ \mathscr{z}(\delta t) \in \{-l_{uv}, 0, l_{uv}\} \right\} \end{aligned}$$

*is not empty, the following properties hold:*

1. *$\delta t_{uv} := \min A_{uv}(t) = \min\{\delta t \in (0,\infty) \mid \mathscr{z}(\delta t) \in \{-l_{uv}, 0, l_{uv}\}\} \in (0,\infty)$ exists.*
2. *$\delta t_{uv}$ is a positive pass-through time, inverse pass-through time, self-loop time or inverse self-loop time. More specifically, $\delta t_{uv}$ is (always w.r.t. $e_{uv}$ and $t$) ...*

$$\begin{aligned} &\dots \textit{ an inverse pass-through time} && \iff \mathscr{z}(\delta t_{uv}) = -l_{uv}, \\ &\dots \textit{ a self-loop time or an inverse self-loop time} && \iff \mathscr{z}(\delta t_{uv}) = 0, \\ &\dots \textit{ a pass-through time} && \iff \mathscr{z}(\delta t_{uv}) = l_{uv}. \end{aligned}$$

3. *There exists no other positive pass-through time, inverse pass-through time, self-loop time or inverse self-loop time than $\delta t_{uv}$.*

Theorem C.13 guarantees that the mapping

$$\mathscr{s} : \mathbb{R} \times E \times F(\mathbb{R}, \mathbb{R}^{n_e}) \longrightarrow (0,\infty),\ (t, e_{uv}, \boldsymbol{\nu}) \longmapsto \mathscr{s}(t, e_{uv}, \boldsymbol{\nu}) := \delta t_{uv} \tag{27}$$

is well-defined. Throughout most parts of this work, $t \in \mathbb{R}$, $e_{uv} \in E$ and $\boldsymbol{\nu} = (\nu_e)_{e \in E}$ are known from the context and we use the notation $\delta t_{uv}$ as a shortcut for $\mathscr{s}(t, e_{uv}, \boldsymbol{\nu})$ to ease notation whenever it is reasonable. Finally, this allows to define the advection-message-generation function in accordance to the results of Theorems C.11 and C.12 as

$$\begin{aligned} &\phi : \mathbb{R} \times E \times F(\mathbb{R}, \mathbb{R}^{n_n}) \times F(\mathbb{R}, \mathbb{R}^{n_e}) \longrightarrow \mathbb{R}_{\geq 0}, \\ &(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu}) \longmapsto \phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu}) := \begin{cases} c_u(t - \delta t_{uv}) & \text{if } \mathscr{z}(\delta t_{uv}) \in \{-l_{uv}, l_{uv}\} \\ c_v(t - \delta t_{uv}) & \text{if } \mathscr{z}(\delta t_{uv}) = 0 \end{cases} \quad \text{(Equation (9))}. \end{aligned}$$

*Remark* C.14 (Double assignment phenomenon). Note that through the relation[9]

$$\int_{t-\delta t_{uv}}^{t} \nu_{vu}(s)\ ds = -\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds,$$

[9]See, e.g., the proof of Lemma C.9.

two neighboring nodes $v \in V$ and $u \in \mathcal{N}(v)$ will be assigned to the same transport time $\delta t_{uv}$, corresponding to either pass-through and inverse pass-through times (cf. Lemma C.9) or self-loop and inverse self-loop times (cf. Lemma C.10). We call this the *double-assignment phenomenon*.

In the following Section C.1.2, we integrate our advection-message-generation function $\phi$ into the coupling conditions (Eq. (4)) and show that the resulting representation of such corresponds to our advection-message-passing algorithm as presented in Section 3 (Eq. (10)) and which automatically handles the double-assignment phenomenon.

### C.1.2 Message Aggregation and Update: Mixing at Nodes

As investigated in the previous Section C.1.1, for a node $v \in V$ and a time $t \in \mathbb{R}$, we are interested in a representation of the advection coupling conditions (Eq. (4)) that does not depend on the concentrations $c_{uv}(t, l_{uv})$ at the end $z = l_{uv}$ of the edges $e_{uv}$ for each $u \in \mathcal{N}_-(v,t)$, but on the concentrations $c_u(t - \delta t_{uv})$ and $c_v(t - \delta t_{uv})$ shifted by suitable *transport times* $\delta t_{uv} \in \mathbb{R}$. Since in practice, we can only assume to have knowledge of the past, the suitability refers to the requirement that $\delta t_{uv} > 0$ should hold. Theorem C.13 guarantees the existence and uniqueness of such.

In this subsection, we prove that the message-generation function as defined in Equation (9) indeed assigns to each time $t \in \mathbb{R}$ and each edge $e_{uv} \in E$ a message $\phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu})$ that determines the correct representation of $c_{uv}(t, l_{uv})$ from the correct source under consideration of the concentrations $\mathbf{c}$ and the flow field $\boldsymbol{\nu}$. The following lemma will help reaching this goal.

**Lemma C.15** (Transport times)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. In the setting of Theorem C.13, the following property holds:*

1. *$\delta t_{uv}$ is (always w.r.t. $e_{uv}$ and $t$) ...*

$$
\begin{aligned}
&\dots \textit{ a pass-through time} && \iff \mathrm{z}(\delta t_{uv}) \in \{-l_{uv}, l_{uv}\} \textit{ and } q_{uv}(t) > 0,\\
&\dots \textit{ an inverse pass-through time} && \iff \mathrm{z}(\delta t_{uv}) \in \{-l_{uv}, l_{uv}\} \textit{ and } q_{uv}(t) < 0,\\
&\dots \textit{ a self-loop time} && \iff \mathrm{z}(\delta t_{uv}) = 0 \textit{ and } q_{uv}(t) < 0,\\
&\dots \textit{ an inverse self-loop time} && \iff \mathrm{z}(\delta t_{uv}) = 0 \textit{ and } q_{uv}(t) > 0.
\end{aligned}
$$

Lemma C.15 justifies why in the definition of the advection-message-generation function $\phi$ (Eq. (9)), we do not need to distinguish between pass-through and inverse pass-through or self-loop and inverse self-loop times and though automatically handle the double assignment phenomenon (cf. Remark C.14): Taking into account the sign of the flow rate $q_{uv}(t)$ at time $t \in \mathbb{R}$, corresponding counterparts will vanish in the coupling condition due to the summation over inflow neighbors $u \in \mathcal{N}_-(v,t)$ only (cf. Eq. (4)) and in our advection-message-passing algorithm due to the ReLU-function (cf. Eq. (6) and (10)).

Moreover, the assumptions of Theorem C.13 and Lemma C.15 are naturally satisfied in the contexts where they are applied: On the one hand, summands where the velocity $\nu_{uv}(t)$ is equal to zero do not contribute by construction of the coupling conditions (4). On the other hand, whenever the set $A_{uv}(t)$ is empty (that is, no transport time exists), we assume to have knowledge of the values $c_v(t)$ for the respective times $t$: In cases of initial value problems, these values are known by definition (cf. Remark 3.8); for boundary value problems we assume an initial state of zero at non-boundary nodes, these values $c_v(t)$ for the respective times $t$ are then also zero by definition.

Consequently, we can combine the results from the advective transport along edges (Section C.1.1) with the results of this Section to obtain a message-passing representation of the coupling conditions (Eq. (4)) which indeed coincides with our final advection-message-passing scheme (10):

**Theorem C.16** (Physical alignment (full version))**.** *Let $v \in V$ and $t \in \mathbb{R}$ hold. We assume that Assumption 3.4 holds and that for each $u \in \mathcal{N}(v)$, $A_{uv}(t)$ is not empty (that is, the OP* (8) *has a solution). If the function $c_v$ obeys the mixing at nodes (Eq.* (4)*) and if for each $u \in \mathcal{N}(v)$, the function $c_{uv}$ obeys the advection edge*

*dynamics (Eq. (2)) with inflow continuity condition (Eq. (5)), then*

$$c_v(t) = \sum_{\substack{u\in\mathcal{N}(v)\\ \text{ʒ}(\delta t_{uv})=\pm l_{uv}}} w_{uv}(t)\cdot c_u(t-\delta t_{uv}) + \sum_{\substack{u\in\mathcal{N}(v)\\ \text{ʒ}(\delta t_{uv})=0}} w_{uv}(t)\cdot c_v(t-\delta t_{uv}) = \sum_{u\in\mathcal{N}(v)} w_{uv}(t)\cdot\varphi(t,e_{uv},\mathbf{c},\boldsymbol{\nu})$$

*holds, where $\delta t_{uv} = \min A_{uv}(t)$ is the transport time as defined in Theorem C.13. In other words, Equation (10) with $\phi$ as defined in Equation (9) holds.*

### C.2 Error Bounds on MeGA-MP

In this section, we first provide some theory on a special graph structure, trees, and introduce further notation along the way (Section C.2.1). This does not only allow to formalize the full version of Theorem 3.7, but also to prove Theorem 3.6 and 3.7 (Section C.2.2). Nevertheless, the following sections can also help to get a deeper understanding of our iterative algorithm, Equation (11), whose intuition was already describe in Section 3.2.

#### C.2.1 Preliminaries: Trees with Children-Aggregation Scheme

In this section, we first investigate on the intuition behind the message passing scheme (10) (and analogously, Eq. (11)), which – by the definition of the weights $w_{uv}$ (Eq. (6)) – can also be written as

$$c_v(t) = \sum_{u\in\mathcal{N}_-(v,t)} w_{uv}(t)\cdot\phi(t,e_{uv},\mathbf{c},\boldsymbol{\nu}).$$

For simplicity, let us assume that the message function $\phi$ outputs $\phi(t,e_{uv},\mathbf{c},\boldsymbol{\nu}) = c_u(t-\delta t_{uv})$ (cf. Eq. (9)), that is, that at time $t$, we observe a pass-through along the edge $e_{uv}$ and not a an inverse self-loop (cf. Lemma C.15). In this case, Equation (10) can – with (1) a change in notation and (2) using the more formal notation from Equation (27) – be written as

$$c_v(t) \overset{(1)}{=} \sum_{v_1\in\mathcal{N}_-(v,t)} w_{v_1v}(t)\cdot c_{v_1}(t-\delta t_{v_1v}) \overset{(2)}{=} \sum_{v_1\in\mathcal{N}_-(v,t)} w_{v_1v}(t)\cdot c_{v_1}(\underbrace{t-\text{ş}(t,e_{v_1v},\boldsymbol{\nu})}_{=:t_1}).$$

For each inflow neighbor $v_1\in\mathcal{N}_-(v,t)$ of $v$ at time $t$, we can again apply Equation (10), yielding

$$c_v(t) = \sum_{v_1\in\mathcal{N}_-(v,t)}\sum_{v_2\in\mathcal{N}_-(v_1,t_1)} w_{v_1v}(t)\cdot w_{v_2v_1}(t_1)\cdot c_{v_2}(\underbrace{t_1-\text{ş}(t_1,e_{v_2v_1},\boldsymbol{\nu})}_{=:t_2}).$$

Intuitively, we can repeat this process as often as we want until a neighborhood $\mathcal{N}_-(v_l,t_l)$ is empty, which by definition is the case if $v_l\in V_\text{b}$ is a source node. What we create along the way are paths $p = ((v_0,t_0) := (v,t),(v_1,t_1),(v_2,t_2),...)$ of both locations $v_l\in V$ and times $t_l\in\mathbb{R}$ for indices $l=1,2,...$ along which we trace back information that contributes to the signal $c_v(t)$.[10] Knowledge of such paths, which we call *inflow paths*, can be helpful in case where the signal $c_v(t)$ at time $t$ is unknown, expressing it through signals $c_{v_l}(t_l)$ at earlier times $t_l<t$. While the graph $\mathcal{G}$ is a symmetric directed graph (cf. Section 2), the set of all such inflow paths $\mathcal{P}(v,t)$ defines a tree structure[11], which we construct formally and iteratively from the root node $(v_0,t_0)=(v,t)$ to leaf nodes in the next Section C.2.2. Before, we provide a general result on trees with children-aggregation aggregation schemes:

**Theorem C.17** (Trees with children-aggregation scheme)**.** *Let $\mathcal{T}=(A,E)$ be a finite tree with root node $a_0\in A$, **int**ernal nodes $A_{int}$ and **leaf** nodes $A_{leaf}$ (that is, $A=\{a_0\}\sqcup A_{int}\sqcup A_{leaf}$ holds). If each node $a\in A$ is associated with a magnitude $x_a\in\mathbb{R}$ that satisfies*

$$x_a = \sum_{b\in\mathcal{C}(a)} \omega_{ab}\cdot x_b \text{ for all } a\in\{a_0\}\sqcup A_{int},$$

[10]Note that the indices $l=1,2,...$ do *not* necessarily correspond to the discrete times indices $I$ in Section 3.2.
[11]Note that by definition, the transport time is positive (Theorem C.13). Therefore, $t>t_1>t_2>...$ holds.

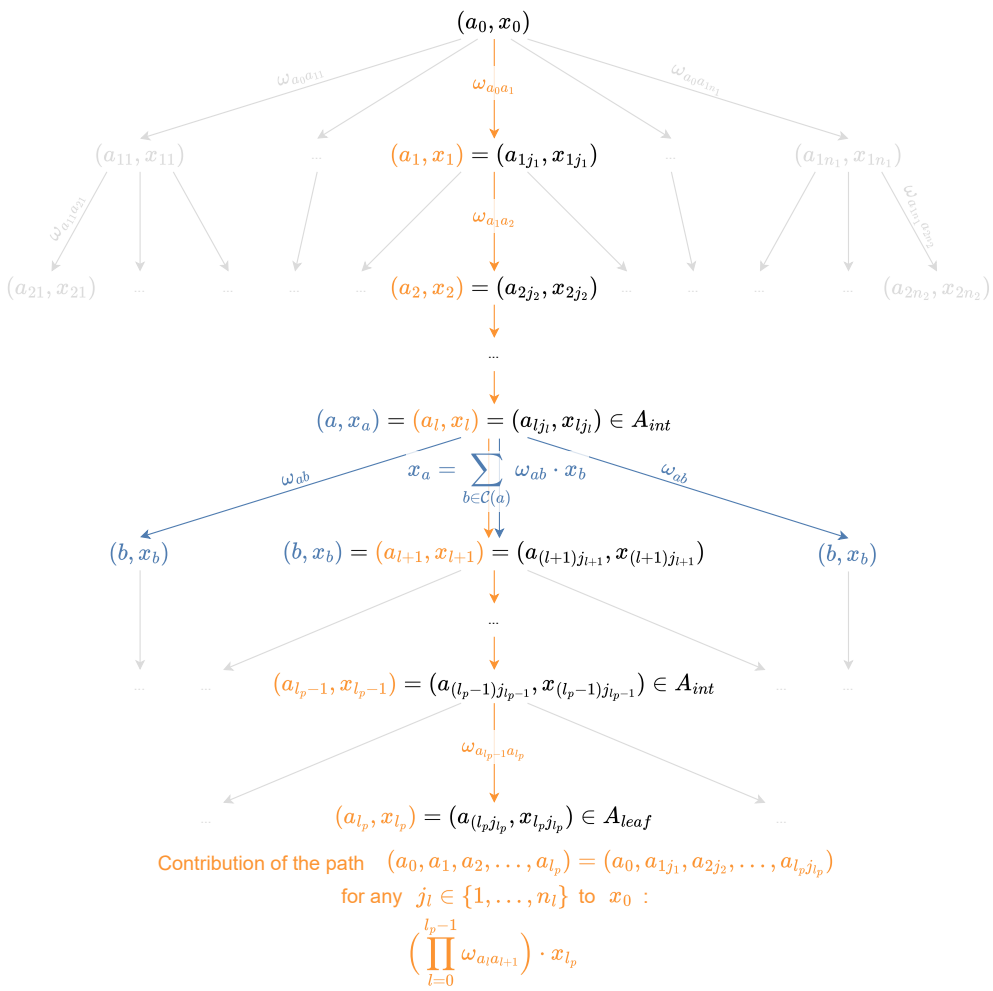



Figure 11: A visualization of the concepts from Theorem C.17.

*where the set $\mathcal{C}(a) \subset A$ consists of the children of $a \in A$ and $w_{ab}$ is a weight of the edge $e_{ab} \in E$ that depends on both the parent $a \in A$ and its child $b \in \mathcal{C}(a)$, then the magnitude $x_0 := x_{a_0}$ of the root node $a_0 \in A$ is determined by the magnitudes $x_{l_p} := x_{a_{l_p}}$ of the leaf nodes $a_{l_p} \in A_{leaf}$ by*

$$x_0 = \sum_{p=(a_0,\ldots,a_{l_p}) \in \mathcal{P}(a_0, A_{leaf})} \omega(a_0, ..., a_{l_p}) \cdot x_{l_p},$$

*where $\mathcal{P}(a_0, A_{leaf})$ is the set of paths connecting the root node $a_0$ to one of its leaf nodes $A_{leaf}$ and*

$$\omega(a_0, ..., a_{l_p}) := \prod_{l=0}^{l_p-1} \omega_{a_l a_{l+1}}$$

*is the product of weights along a(n) (arbitrary) path $p = (a_0, ..., a_{l_p}) \in \mathcal{P}$ with length $l_p \in \mathbb{N}$.*

A visualization of such a tree can be found in Figure 11.

### C.2.2 Construction of Inflow Trees

In this section, we first provide some intermediate results – Theorem C.20 and Theorem C.21 – that help proving Theorem 3.6 and 3.7. In order to prove the former, we construct suitable trees in the sense of Theorem C.17 as already motivated in Section C.2.1:

**Definition C.18** (Inflow tree and interpolation inflow tree)**.** Let $\mathcal{G} = (V, E)$ be an advection metric graph as defined in Section 2. Let $v \in V$ and $t \in \mathbb{R}$ hold.

We define the inflow tree $\mathcal{T}(v, t)$ of $v$ at time $t$ using the notation from Theorem C.17 by iteratively defining the nodes and children per layer $l \in \mathbb{N}_0$ of the tree until a stopping criteria is fulfilled:

- $l = 0$-th layer: We choose the root node $a_0 = (v_0, t_0) := (v, t)$.
- $l + 1$-th layer: Given a parent node $a_l = (u_l, v_l, t_l)$[12] in the $l$-th layer, we choose its children $a_{l+1} = (u_{l+1}, v_{l+1}, t_{l+1}) \in \mathcal{C}(a_l)$ as each combination of
  - an inflow neighbor $u_{l+1} \in \mathcal{N}_-(v_l, t_l)$ of $v_l$ at time $t_l$ and
  - the corresponding shifted time $t_{l+1} := t_l - \mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})$ obtained through the time shift applied to $t_l$ using the (positive) transport time $\mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu}) > 0$ (cf. Eq. (27)) along the edge $e_{u_{l+1}v_l} \in E$ and
  - the corresponding source node[13]

$$v_{l+1} := \begin{cases} u_{l+1} & \text{if } \mathfrak{z}(\mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})) \neq 0, \\ v_l & \text{if } \mathfrak{z}(\mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})) = 0. \end{cases} \tag{28}$$

- We stop the construction after an $a_{l+1} = (u_{l+1}, v_{l+1}, t_{l+1})$ if a stopping criteria is fulfilled, which for now can be any stopping criteria, or until naturally, the neighborhood $\mathcal{N}_-(v_{l+1}, t_{l+1})$ is empty, which is by definition the case if $v_{l+1} \in V_\mathrm{b}$ is a source node.
- We denote the set of paths connecting the root node $a_0 = (v_0, t_0) = (v, v, t)$ to one of its leaf nodes $A_\mathrm{leaf}$ by $\mathcal{P}(v, t)$, which is an abbreviation for $\mathcal{P}(a_0, A_\mathrm{leaf})$.

We similarly define the interpolation inflow tree $\hat{\mathcal{T}}(v, t)$ of $v$ at time $t$ using the notation from Theorem C.17 by iteratively defining the nodes and children per layer $l \in \mathbb{N}_0$ of the tree until a stopping criteria is fulfilled:

- $l = 0$-th layer: We choose the root node $a_0 = (v_0, t_0) := (v, t)$.
- $l + 1$-th layer: Given a parent node $a_l = (u_l, v_l, t_l)$ in the $l$-th layer, we choose its children $a_{l+1} = (u_{l+1}, v_{l+1}, t_{l+1}) \in \mathcal{C}(a_l)$ as each combination of
  - an inflow neighbor $u_{l+1} \in \mathcal{N}_-(v_l, t_l)$ of $v_l$ at time $t_l$ and
  - an corresponding interpolation time $t_{l+1} \in \{t_j, t_{j+1}\}$ for the $j \in I$ for which $t_l - \mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu}) \in [t_j, t_{j+1})$ holds (cf. Eq. (17))[14] and
  - the corresponding source node $v_{l+1}$ analogously to Equation (28).[15]

[12] Note that the definition of $a_{l+1}$ does not depend on $u_l$, which is why this step is well-defined for $l = 0$ and $l + 1 = 1$, too.

[13] In the former case, we observe a pass-through along the edge $e_{u_{l+1}v_l}$, that is, information flows from $v_{l+1} = u_{l+1}$ to $v_l$. In the latter case, we observe an inverse self-loop along the edge $e_{u_{l+1}v_l}$, that is, information flows from $v_{l+1} = v_l$ to $v_l$. For more details, see Section C.1.

[14] Similar to the construction to the inflow tree $\mathcal{T}(v, t)$, $t_l - \mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})$ is the – possibly not in the observable, discrete subdomain of $\mathbb{R}$ lying – shifted time obtained through the time shift applied to $t_l$ using the (positive) transport time $\mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu}) > 0$ (cf. Eq. (27)) along the edge $e_{u_{l+1}v_l} \in E$. Note that for $l \in \mathbb{N}$, $t_l$ already corresponds to one of the discrete time points that are known, that is, there exists an $j \in I$ such that $t_l = t_j$ holds.

[15] Note that the definition of $v_{l+1}$ does not depend on $t_{l+1}$, which is why the corresponding source node $v_{l+1}$ of both $t_{l+1} = t_j$ and $t_{l+1} = t_{j+1}$ is the same.

- We stop the construction after an $a_{l+1} = (u_{l+1}, v_{l+1}, t_{l+1})$ for which $t_{l+1} \in T_{\text{hi}} = \{t_1, ..., t_i\}$ or $t_{l+1} \in T_{\text{fu}} = \{t_{i+1}, ..., t_n\}$ and $v_{l+1} \in V_{\text{b}}$ holds.[16]
- We denote the set of paths connecting the root node $a_0 = (v_0, t_0) = (v, t)$ to one of its leaf nodes $A_{\text{leaf}}$ by $\hat{\mathcal{P}}(v, t)$, which is an abbreviation for $\hat{\mathcal{P}}(a_0, A_{\text{leaf}})$.

*Remark* C.19 (Finite depths of interpolation inflow tree). We want to explicitly highlight that for each $v \in V$ and $t \in \mathbb{R}$, the stopping criteria for the interpolation inflow tree $\hat{\mathcal{T}}(v, t)$ is fulfilled at some point. In other words, the interpolation inflow tree $\hat{\mathcal{T}}(v, t)$ has a finite depth and each path $p = ((v_0, t_0) = (v, t), (u_1, v_1, t_1)..., (u_{l_p}, v_{l_p}, t_{l_p})) \in \hat{\mathcal{P}}(v, t)$ has a finite length $l_p \in \mathbb{N}$. This is because by definition of the OP (8), we assume positive transport times (also see discussion above Theorem C.16). Consequently, $t = t_0 > t_1 > t_2 > ...$ holds and for arbitrary $t \in \mathbb{R}$, at some point, $t_{l_p} \in T_{\text{hi}}$ must hold.

We now leverage these trees together with Theorem C.17 to prove the following two intermediate results, which allow to reconstruct the magnitudes $c_v(t)$ and $\hat{c}_v^{(k)}(t)$ of interest along inflow paths:

**Theorem C.20** (Closed form of advection message passing)**.** *Let $v \in V$ and $t \in \mathbb{R}$ hold. Let $\mathcal{T}(v, t)$ be the inflow tree of $v$ at time $t$ as defined in Definition C.18. If $c_v(t)$ is defined by Equation* (10)*, then*

$$c_v(t) = \sum_{p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)...,(u_{l_p},v_{l_p},t_{l_p}))\in\mathcal{P}(v,t)} \left( \prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l) \right) \cdot c_{v_{l_p}}(t_{l_p}) \tag{29}$$

*with weights $w_{u_{l+1}v_l}(t_l)$ according to Equation* (6) *holds.*

**Theorem C.21** (Closed form of advection message passing iterations)**.** *Let $v \in V$ and $t \in T_{hi} \cup T_{fu}$ hold. Let $\hat{\mathcal{T}}(v, t)$ be the interpolation inflow tree of $v$ at time $t$ as defined in Definition C.18. Let $k \geq \max_{p\in\hat{\mathcal{P}}(v,t)} l_p$ hold. If $\hat{c}_v^{(k)}(t)$ is defined by Equation* (11)*, then*

$$\hat{c}_v^{(k)}(t) = \sum_{p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)...,(u_{l_p},v_{l_p},t_{l_p}))\in\hat{\mathcal{P}}(v,t)} \left( \prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l) \cdot a(t_{l+1}) \right) \cdot \hat{c}_{v_{l_p}}^{(k-l_p)}(t_{l_p}) \tag{30}$$

*with weights $w_{u_{l+1}v_l}(t_l)$ and $a(t_{l+1})$ according to Equation* (6) *and* (16)*, respectively, holds.*

Theorem C.21 already allows a proof of Theorem 3.6, which is given together with the other proofs in Appendix E.

For the proof of Theorem 3.7, we need to make some further assumptions: While Equation (29) and (30) are already of similar structure, the number of paths $\hat{p} \in \hat{\mathcal{P}}(v, t)$ grows exponentially by a factor of $2^l$, where $l$ is the depth of the tree $T(v, t)$ and $\hat{\mathcal{T}}(v, t)$, as compared to the number of paths $p \in P(v, t)$. This is due the fact that per time that we observe in the inflow tree $T(v, t)$, we observe two interpolation times in the interpolation inflow tree $\hat{\mathcal{T}}(v, t)$ (cf. Definition C.18). To make each summand in Equation (30) comparable to a summand in Equation (29), we need to make the following assumptions:

**Assumption C.22** (Identifiable paths)**.** Let $v \in V$ and $t \in \mathbb{R}$ hold. Let $T(v, t)$ and $\hat{\mathcal{T}}(v, t)$ be the inflow tree and the interpolation inflow tree from Definition C.18, respectively. We assume that each path $\hat{p} = ((\hat{v}_0, \hat{t}_0) = (v, t), (\hat{u}_1, \hat{v}_1, \hat{t}_1)..., (\hat{u}_{l_p}, \hat{v}_{l_p}, \hat{t}_{l_p})) \in \hat{\mathcal{P}}(v, t)$ is uniquely identifiable with a path $p = ((v_0, t_0) = (v, t), (u_1, v_1, t_1)..., (u_{l_p}, v_{l_p}, t_{l_p})) \in \mathcal{P}(v, t)$ in the sense that $\hat{u}_l = u_l$ and $\hat{v}_l = v_l$ holds for all $l = 1, ..., l_p$.

**Assumption C.23** (Equal time for jumping into history)**.** Let $v \in V$ and $t \in \mathbb{R}$ hold. Let $\hat{\mathcal{T}}(v, t)$ be the interpolation inflow tree from Definition C.18. In the definition of each $t_{l+1} \in \{t_j, t_{j+1}\}$, we assume that $t_j, t_{j+1} \in T_{\text{fu}}$ or $t_j, t_{j+1} \in T_{\text{hi}}$ holds.

[16]Note that similar to the construction to the inflow tree $\mathcal{T}(v, t)$, we can also chose any other suitable stopping criteria. We chose the one that we will need later. Also note that with this stopping criterium, if $t \in T_{\text{hi}}$ or $t \in T_{\text{fu}}$ and $v \in V_{\text{b}}$ holds, the creation of the tree $\hat{\mathcal{T}}(v, t)$ stops immediately and does only consist of the path $p = ((v_0, t_0) = (v, t))$ of length zero.

Assumption C.22 is satisfied as long as the inflow neighborhoods $\mathcal{N}_-(v_l, t_l)$, from which we choose an inflow neighbor $u_{l+1}$ when creating a path in $\mathcal{T}(v,t)$, are equal to the corresponding neighborhoods $\mathcal{N}_-(\hat{v}_l, \hat{t}_l)$, from which we choose an inflow neighbor $\hat{u}_{l+1}$ when creating a path in $\hat{\mathcal{T}}(v,t)$. This is guaranteed if the underlying flow field $\boldsymbol{\nu}$ that defines these neighborhoods does not change its orientation during the time intervals $[t_j, t_{j+1}]$ for which $t_l \in [t_j, t_{j+1})$ holds. If the assumption holds, it allows the following definition.

**Definition C.24** (Joint paths)**.** Let $v \in V$ and $t \in \mathbb{R}$ hold. Let $T(v,t)$ and $\hat{\mathcal{T}}(v,t)$ be the inflow tree and the interpolation inflow tree from Definition C.18, respectively. Let assumption C.22 hold. If the path $\hat{p} = ((\hat{v}_0, \hat{t}_0) = (v,t), (\hat{u}_1, \hat{v}_1, \hat{t}_1)..., (\hat{u}_{l_p}, \hat{v}_{l_p}, \hat{t}_{l_p})) \in \hat{\mathcal{P}}(v,t)$ is uniquely identifiable with the path $p = ((v_0, t_0) = (v,t), (u_1, v_1, t_1)..., (u_{l_p}, v_{l_p}, t_{l_p})) \in \mathcal{P}(v,t)$, we define the joint path of $\hat{p}$ by

$$\tilde{p} = ((v_0, t_0) = (v,t), (u_1, v_1, t_1, \hat{t}_1), ..., (u_{l_p}, v_{l_p}, t_{l_p}, \hat{t}_{l_p})). \tag{31}$$

We can still consider $\tilde{p}$ as an element of $\hat{\mathcal{T}}(v,t)$, since we only added information that are unused, in, e.g., Theorem C.21.

Moreover, Assumption C.23 guarantees that there exist two such paths that are exactly equal up to the last $\hat{t}_{l_p} \in \{t_j, t_{j+1}\}$ for a $j \in I$ in the sense of Definition C.18. Finally, using the property that per such two paths ending in such two left and right interpolation bounds, the factors $a(t_j)$ and $a(t_{j+1})$ add up to one, we can canonically extend the sum in Equation (29) to the sum in Equation (30) using the joint paths. This allows to prove Theorem 3.7, which in its full version is phrased as follows:

**Theorem C.25** (Interpolation error (full version))**.** *Let $v \in V$ and $t \in T_{hi} \cup T_{fu}$ hold. Let $\mathcal{T}(v,t)$ and $\hat{\mathcal{T}}(v,t)$ be the inflow tree and the interpolation inflow tree of $v$ at time $t$ as defined in Definition C.18, respectively. Let Assumption C.22 and C.23 hold. Then the prediction error in-between $c_v(t)$ and $\hat{c}_v(t)$ is given by*

$$\begin{aligned} &|c_v(t) - \hat{c}_v(t)| \\ \leq &\sum_{\tilde{p} \in \hat{\mathcal{P}}(v,t)} a(\hat{t}_1, ..., \hat{t}_{l_p}) \cdot \left( \underbrace{|c_{v_{l_p}}(t_{l_p}) - c_{v_{l_p}}(\hat{t}_{l_p})|}_{\textit{Temporal interpolation error}} + |c_{v_{l_p}}(t_{l_p})| \cdot \underbrace{|w(t_0, ..., t_{l_p-1}) - w(\hat{t}_0, ..., \hat{t}_{l_p-1})|}_{\textit{Accumulated weighting error}} \right) \end{aligned}$$

*with joint paths $\tilde{p} = ((v_0, t_0) = (v,t), (u_1, v_1, t_1, \hat{t}_1), ..., (u_{l_p}, v_{l_p}, t_{l_p}, \hat{t}_{l_p})) \in \hat{\mathcal{P}}(v,t)$ according to Definition C.24,*

$$a(\hat{t}_1, ..., \hat{t}_{l_p}) := \prod_{l=0}^{l_p-1} a(\hat{t}_{l+1}) \quad \textit{and} \quad w(\hat{t}_0, ..., \hat{t}_{l_p-1}) := \prod_{l=0}^{l_p-1} w_{u_{l+1} v_l}(\hat{t}_l)$$

*with weights $w_{u_{l+1} v_l}(t_l)$ and $a(t_{l+1})$ according to Equation* (6) *and* (16)*, respectively. If the temporal interpolation error is zero, we obtain $|c_v(t) - \hat{c}_v(t)| = 0$.*

Theorem C.25 states that the error between the ground truth $c_v(t)$ and the prediction $\hat{c}_v(t)$ solely stems from the time interpolation: On the one hand, the temporal interpolation error is the error from observing a value $c_{v_{l_p}}(\hat{t}_{l_p})$ at a node $v_{l_p} \in V$ and at a discrete time $t_{l_p} \in T_{\text{hi}}$, when actually, we needed to observe the value $c_{v_{l_p}}(t_{l_p})$ at an eventually non-discrete time $t_{l_p} \in \mathbb{R}$. On the other hand, for exactly the same reason, the weights $w_{u_{l+1} v_l}(\hat{t}_l)$ that we observe at discrete times $\hat{t}_l \in T_{\text{hi}} \sqcup T_{\text{fu}}$ might differ from the weights $w_{u_{l+1} v_l}(t_l)$ that we needed to observe at non-discrete times $t_l \in \mathbb{R}$ for all $l = 1, ..., l_p - 1$, yielding the accumulated weighted error. Both errors decrease with decreasing content of high frequencies in concentrations $\mathbf{c}$ and flow fields $\boldsymbol{\nu}$.

# D Details on the Experiments

## D.1 Experiment 1: Zero-Shot Transfer Learning of Advection-Reaction Dynamics on Real-World Metric Graphs

**Domain** In both networks Hanoi and L-Town, we model chlorine decay, which is part of the edge dynamics in Equation (12), as a first-order reaction over time and also include interactions with the pipe wall, which

depend on the pipe diameter $\rho_{uv} \in \mathbb{R}_{\geq 0}$. This is formally defined as

$$f_r(t, z, e_{uv}) = -\left(k_b + \frac{4k_w}{\rho_{uv}}\right) c_{uv}(t, z), \tag{32}$$

where the parameters $k_b \in \mathbb{R}_{\geq 0}$ and $k_w \in \mathbb{R}_{\geq 0}$ control the intensity of chlorine decay over time and due to wall reactions, respectively.

**Dataset** We generate datasets of 1000 training, 200 validation, and 200 testing samples for the Hanoi WDS, and 200 testing samples for the L-Town WDS. Each sample consists of $n = 701$ time steps with a time difference of $dt = 60$ (in s) and corresponds to a time series $T$ that describes the distribution of chlorine $\mathbf{c}$ over discrete time points $t \in T$ and through the WDS $\mathcal{G}$, determined by advection or advection-reaction dynamics as described in section 4.1. The chlorine is injected over time $t \in T$ at boundary nodes $V_\mathrm{b}$. The samples differ in graph topology, in flow fields and injection pattern (boundary condition). We now describe this data generation process in more detail. As stated in Section 4, we use EPyT-Flow (Artelt et al., 2024) based on EPANET-MSX (Shang et al., 2023) as the simulator.

- **Fixed parameters:** Boundary nodes $V_\mathrm{b}$ at which chlorine is injected correspond to reservoir nodes $V_\mathrm{r}$. This corresponds to one and two boundary nodes in the case on Hanoi and L-Town, respectively. For the advection-reaction datasets, the values of the reaction coefficients in Equation (32) that model the decay of the concentration are set to $k_b = 0.04$ and $k_w = 0.034$.
- **Varying graph topology:** In order to vary graph topology, we sample pipe lengths $l_{uv} \sim U(0.1, 80)$ (in m) and pipe diameters $\rho_{uv} \sim U(0.06, 0.025)$ (in m) uniformly per sample at random. To maintain the typical WDS topology of larger pipes at reservoirs and successively smaller pipes towards leaf nodes, diameters are ordered from low to high, as in the original WDSs Hanoi and L-Town.
- **Varying flow fields:** In order to generate scenarios with different flow fields, we vary nodal demands $\mathbf{d}(t) = (d_v(t))_{v \in V}$ that directly determine the flow field $\boldsymbol{\nu}(t) = (\nu_e(t))_{e \in E}$ through hydraulic equations (Rossman et al., 2020). Empirically, independent random demands yield spatially homogeneous flow fields. To introduce spatial variations of the flow fields, we apply spectral clustering $s : V \to \{0, ..., n_\mathrm{c}\}$ on the binary network adjacency matrix to partition the network into $n_\mathrm{c} = 4$ clusters for Hanoi and $n_\mathrm{c} = 7$ clusters for L-Town. Then the demand pattern for a node $v \in V$ is sampled from a cluster-shared Gaussian process plus independent noise:

  $$d_v(t) = f_{s(v)}(t) + \epsilon_v, \quad \epsilon_v \sim \mathcal{N}(0, 0.1^2), \quad f_{s(v)} \sim \mathcal{GP}(0, K), \quad K(i, j) = \exp\left(-\frac{(i-j)^2}{2 \cdot 0.1^2 \cdot (n-1)^2}\right).$$

  The resulting hydraulics show flow velocities $\boldsymbol{\nu}$ that vary in space and time and can also change direction. Assigning demands $d_v(t)$ may result in physically implausible states that generate negative pressures, however, these states can be easily made plausible by offsetting the heads accordingly, which would have no effect on the flows. Therefore, the flows are physically plausible and since the quality simulation only depends on flows and not on pressures, the resulting evolution of chlorine in the system is still valid.
- **Varying injection patterns:** The injection patterns $c_v(t)$ (Dirichlet boundary condition) in which chlorine is injected at boundary nodes $v \in V_\mathrm{b}$ are sampled from the Fourier function space (using the first $M$ modes) as follows:

  $$\bar{c}_v(t) = \sum_{k=1}^{M} \frac{1}{k^\alpha} \left(a_k \cos\left(2k\pi \frac{t}{60 \cdot 700}\right) + b_k \sin\left(2k\pi \frac{t}{60 \cdot 700}\right)\right),$$
  $$c_v(t) = \frac{\bar{c}_v(t) - \min_{t \in T} \hat{c}_v(t)}{\max_{t \in T} \bar{c}_v(t) - \min_{t \in T} \bar{c}_v(t)} \in [0, 1],$$

  with $M \sim \mathcal{U}_{int}(3, 30)$, $\alpha \sim \mathcal{U}(0.5, 1.5)$, and $a_k, b_k \sim \mathcal{N}(0, 1)$, for all $v \in V_\mathrm{b}$, $t \in T$. Here $\mathcal{U}_{int}$ corresponds to an integer-valued uniform distribution. Three samples from this function space are visualized in Figure 12.

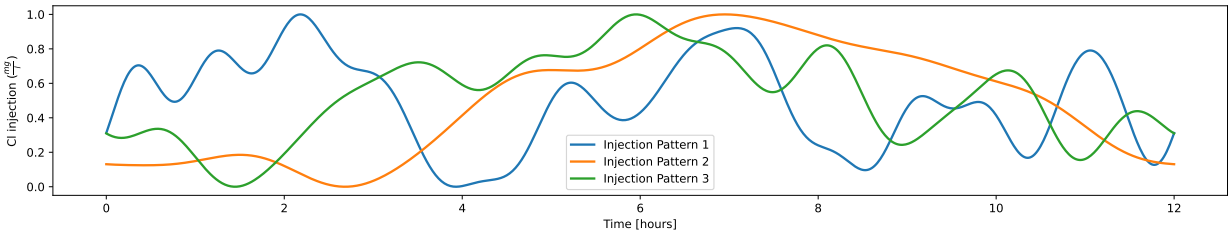



Figure 12: Example patterns used for the chlorine injection at boundary nodes $V_{\mathrm{b}}$.

**Model** The MLP that is intended to learn the reactions consists of two linear layers with 8 neurons each, and a SELU activation (Klambauer et al., 2017) in between. We train the model with the Adam optimizer (Kingma & Ba, 2014) over 250 epochs with a batch size of 256 using a learning rate of 0.001 that is reduced by a factor of 0.2 if the training loss did not decrease for the last 3 epochs. The loss function is the MAE. Boundary nodes are masked.

**Baselines** The baselines we compare ourselves to are:

- **NNConv:** As a simple baseline, we use the kernel-based MPNN from Gilmer et al. (2017). The node feature matrix $\mathbf{X} \in \mathbb{R}^{|V| \times n}$ with $n = n_{\mathrm{hi}} + n_{\mathrm{fu}}$ is populated with the complete time series at boundary nodes $V_{\mathrm{b}}$, as well as the first time step as the known history $T_{\mathrm{hi}}$ at all other nodes $V \setminus V_{\mathrm{b}}$. The edge feature matrix $\mathbf{E} \in \mathbb{R}^{|E| \times 3n}$ contains the time series of scaled edge flows, edge capacities, and transport times. Node and edge features are first mapped into a latent space via two separate linear layers. After that, GG-NN convolutional layers (Gilmer et al., 2017) (NNConv implementation from torch-geometric) are applied. The message function is implemented as a two-layer MLP with a ReLU activation between the two layers. The messages are aggregated via sum aggregation and decoded by a single linear layer calibrated manually.

- **PDE-GNN:** Hermes et al. (2025) is a PDE-style GNN which solves initial and boundary value problems and has been proposed for chlorine estimation in WDSs. The model learns a PDE function that is integrated by Euler integration. Calibrated manually, we apply the same scheduler with 0.2 learning rate decay upon training loss plateau over 3 epochs and we set the initial learning rate to 0.001. The time step used for Euler integration is set to $dt = 0.25$. Some hyperparameters are tuned, other hyperparameters are left as specified in Hermes et al. (2025).

- **GPSConv:** Rampášek et al. (2022) is a graph transformer that we include as a representative of an architecture capable of capturing global dependencies. As positional encodings, we utilize the graph Laplacian eigenvectors, since positional encodings based on random walks would degenerate to zero if the initial conditions are zero. The Laplacian positional encoding is truncated to 30 components, allowing the transfer-learning experiment from Hanoi (32 nodes) to the L-Town (784 nodes), and embedded to 8 dimensions via a linear layer. We tune the message-passing module, which corresponds to the parameter $\Phi$ in Table 4, candidates are NNConv, GINE, and GatedGCN (Gilmer et al., 2017; Hu* et al., 2020; Bresson & Laurent, 2017), which are architectures that can accommodate edge features.

- **A-DGN:** Gravina et al. (2023) is a dynamical-systems based GNN, where the message-passing operation acts as a differential operator, similar to an ODE. As such, it computes updates $\frac{\partial \mathbf{X}}{\partial l}$ of the node features $\mathbf{X}$ that are integrated over several layers $l$ via Euler integration with a step size of $\epsilon$. This model allows deep architectures that are stable and non-dissipative, mediated through antisymmetric weight matrices applied to the output of any message passing layer $\Phi$. We use this model in the weight-sharing style and select the same options for $\Phi$ as for GPSConv. As generalization is a focus here, we evaluate two variations A-DGN and A-DGN$_{Dia}$, to account for the difference in size of the training and evaluation graph: The first is a hyperparameter tuned baseline, while for the second variant, we change the number of iterations of the trained model to the graph

diameter. We have found that the latter works much better when the step size $\epsilon$ is adapted as well. Specifically, we set $\epsilon' = \epsilon \frac{num_iters}{diameter(\mathcal{G})}$, and $num_iters' = diameter(\mathcal{G})$.

- **PINN-based approaches** on metric graphs (Blechschmidt et al., 2022; Laczkó et al., 2025) are effective for learning solutions of specific PDE settings on metric graphs and are suited for inverse problems. However, PINNs are not operator-learning frameworks: A change in PDE parameters or boundary conditions typically requires retraining. Since our focus is on flexible models that generalize across varying flow conditions and topologies, PINN-based approaches are not an appropriate baseline in our setting.
- **PI DEEPONET** is a *LEGO*-like neural operator for metric graphs (Blechschmidt et al., 2025) originally developed for drift–diffusion edge dynamics. While this approach is highly effective for diffusive dynamics, its direct application to purely hyperbolic problems such as linear advection is not straightforward. First, the method assumes flow-directed edges, effectively imposing a fixed transport direction that is incompatible with more general advection fields that can change direction. Second, at inner nodes the coupling is enforced through Kirchhoff–Neumann conditions, which conserve diffusive gradient fluxes $\partial_z c$. For hyperbolic dynamics, the flux is defined as $\nu(t, z)c(t, z)$ which would require a different coupling law such as mixing. For these reasons we omit this work as a baseline for now.

All models are trained using the Adam optimizer over a maximum of 1000 epochs. During training we validate the model every 10 epochs and training is stopped early if the validation loss did not improve over the last 50 epochs.

For a fair comparison of MeGA-MP and MeGA-MP$^+$ to the other methods, we run a small-scale hyperparameter tuning for every baseline method using Optuna (Akiba et al., 2019). The search space is specified in Table 4. Every configuration is trained for 60 epochs. Trials are terminated at an earlier epoch if the trial's result is worse than the median of results of previous trials at the same epoch (median pruning). New configurations are sampled based on the tree-structured Parzen Estimator (TPE) (Watanabe, 2025) implemented in Optuna.

Table 4: Optuna Search Space for Different Model Types

| **Parameter** | **NNConv** | **PDE-GNN** | **GPSConv** | **A-DGN** |
|---|---|---|---|---|
| *hidden_dim* | $\{64, 96, 128, 160\}$ | | | |
| *num_layers* | $\{2, 3, 4, 5, 6\}$ | | | |
| *learning_rate* | $[10^{-4}, 3 \times 10^{-3}]$ (log) | | | |
| *weight_decay* | $[10^{-6}, 10^{-1}]$ (log) | | | |
| *batch_size* | $\{16, 32, 64\}$ (tests with larger batch-sizes were not successful) | | | |
| *clip_grad_norm* | – | $\{10^{-4}, 0.5, 1.0, 2.0, -\}$ | – | – |
| $n_{\text{fu}}$ (training) | 700 | $\{32, 100, 150\}$ | 700 | 700 |
| *dropout* | – | [0.0, 0.3] | [0.0, 0.3] | – |
| *num_heads* | – | – | {2,4,8} | – |
| *attn_dropout* | – | – | [0.0, 0.3] | – |
| *step_size* / $\epsilon$ | – | {1/4, 1/3, 1/2, 1} | – | { 1.0, 0.1, 0.01, 0.001 } |
| $\gamma$ | – | – | – | { 1.0, 0.1, 0.01, 0.001 } |
| $\Phi$ | – | – | { NNConv, GINE, GatedGCN } | { NNConv, GINE, GatedGCN } |
| *num_iters* | – | – | – | { 1, 2, 5, 10, 20, 50 } |
| | **Fixed Hyperparameters** | | | |
| $n_{\text{hi}}$ | 1 | | | |
| $n_{\text{fu}}$ (evaluation) | 700 | | | |
| *loss_fn* | L1 | | | |
| *epochs* | 60 (hyperparameter tuning) / 1000 (training, with early stopping) | | | |

**Results** As an extension of our ablation studies in Section 4.3, in Figures 13 to 16, we compare the performance of MeGA-MP, MeGA-MP$^+$ and MeGA-MP$^-$ on both the advection and the advection-reaction dynamical systems as well as on Hanoi and L-Town qualitatively.

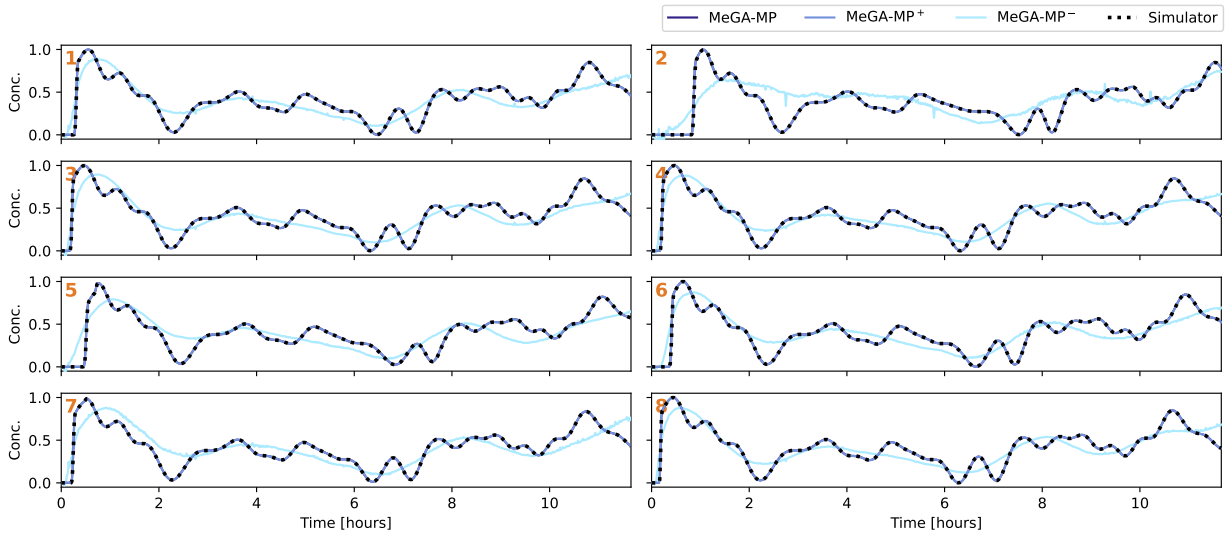



Figure 13: Predictions of MeGA-MP, MeGA-MP$^+$ and MeGA-MP$^-$ for the **advection** dynamical system on the WDS **Hanoi**. The selected nodes are the same as in Figure 5.

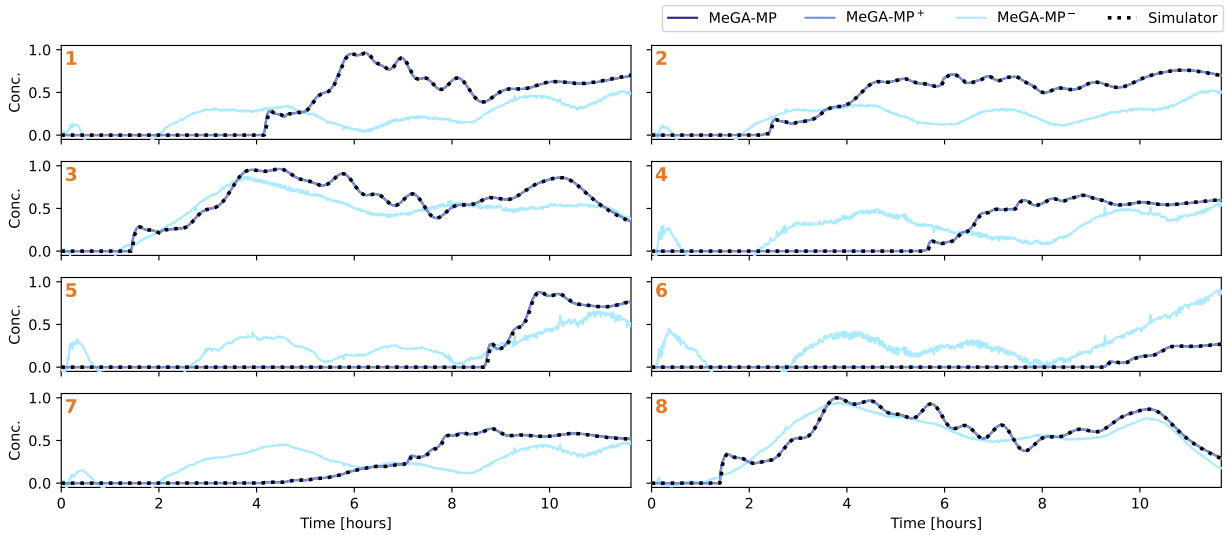



Figure 14: Predictions of MeGA-MP, MeGA-MP$^+$ and MeGA-MP$^-$ for the **advection** dynamical system on the WDS **L-Town**. The selected nodes are the same as in Figure 6.

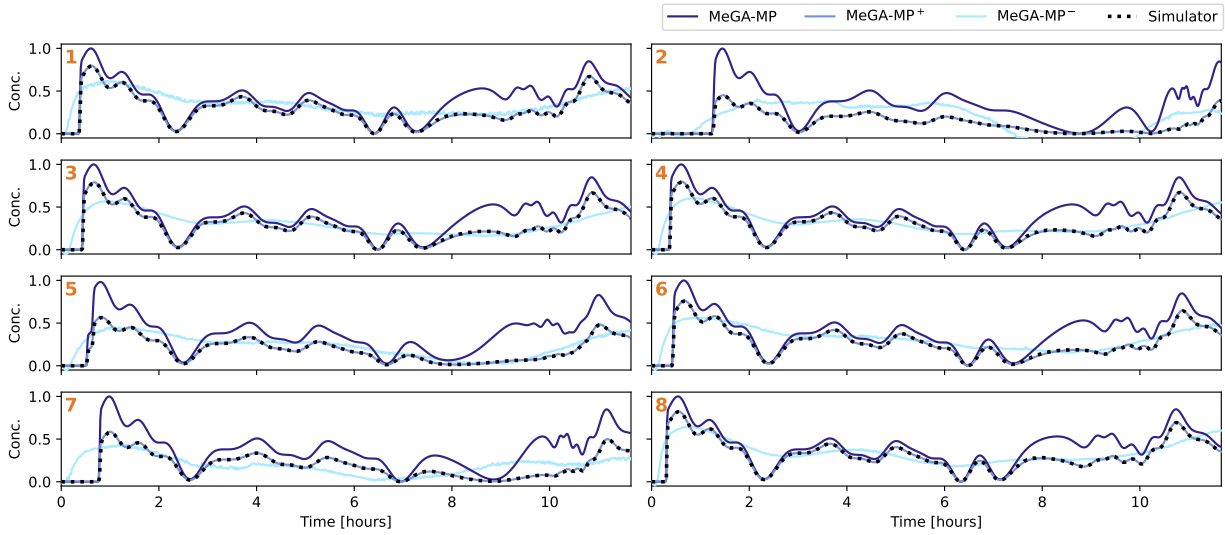



Figure 15: Predictions of MeGA-MP, MeGA-MP$^+$ and MeGA-MP$^-$ for the **advection-reaction** dynamical system on the WDS **Hanoi**. The selected nodes are the same as in Figure 5.

### D.2 Experiment 2: Advection on a 1D Euclidean Domain

**Domain and Dataset** The boundary value problem which we analyse in Section 5 is given by

$$\begin{aligned} \frac{\partial c(t,z)}{\partial t} &= -\nu(t)\,\frac{\partial c(t,z)}{\partial z} \quad \text{with} \quad \nu(t) = 0.3 \cdot \sin\left(\frac{2\pi t}{100}\right) + 0.3 \text{ and} \\ c(t,0) &= \exp\left(\frac{1}{2}\left(\frac{t-\mu_t}{\sigma_t}\right)^2\right) \end{aligned} \tag{33}$$

for all $t \in [0, 100]$ and $z \in [0, 100]$ and with $\mu_t = 38$ and $\sigma_t = 18$. Its analytical solution is computed analogously to the solution of the initial value problem with boundary condition, as discussed next.

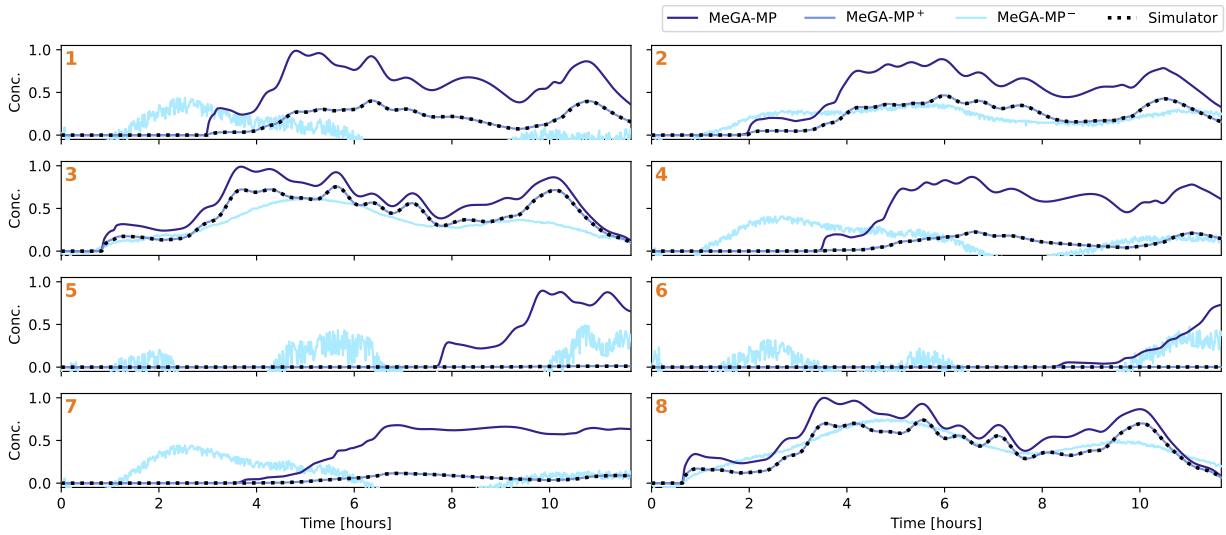



Figure 16: Predictions of MeGA-MP, MeGA-MP$^+$ and MeGA-MP$^-$ for the **advection-reaction** dynamical system on the WDS **L-Town**. The selected nodes are the same as in Figure 6.

Additionally, in this Appendix, we analyze the initial value problem with boundary condition given by

$$\begin{aligned} \frac{\partial c(t,z)}{\partial t} &= -\nu(t)\,\frac{\partial c(t,z)}{\partial z} \quad \text{with} \quad \nu(t) = 0.3 \cdot \sin\left(\frac{2\pi t}{100}\right) + 0.3 \text{ and} \\ c(0,z) &= \exp\left(\frac{1}{2}\left(\frac{z-\mu_z}{\sigma_z}\right)^2\right), \quad c(t,0) = \exp\left(\frac{1}{2}\left(\frac{t-\mu_t}{\sigma_t}\right)^2\right) \end{aligned} \tag{34}$$

for all $t \in [0, 100]$ and $z \in [0, 100]$ and with $\mu_z = 8$, $\sigma_z = 2.5$, $\mu_t = 38$ and $\sigma_t = 18$.

To compute an analytical solution, we can solve the initial value problem $c_{ivp}(t, z)$ and the boundary value problem $c_{bvp}(t, z)$ separately and sum the individual results. This decomposition is possible since for a positive and incompressible flow $\nu(t) > 0$, characteristic lines do not cross and each point of the solution corresponds to a value of either the initial value function or the boundary value function.

The initial value problem is solved by integrating the velocity field to compute the spatial shift of the Gaussian $c(0, z)$ for each $t \in [0, 100]$, via $t \longmapsto \not{z}_0(t) := \int_0^t \nu(s)\, ds$. The antiderivative of $\nu(t)$ is given by

$$\bar{\nu}(t) = 0.3\left(t - \frac{\cos(4\pi t)}{4\pi} + \frac{1}{4\pi}\right), \tag{35}$$

and the solution of the initial value problem is $c_{ivp}(t, z) = c(0, z + \not{z}_0(t))$.

To solve the boundary value problem, we find the transport time $\delta t(z)$ from the boundary at $z = 0$ to every other point $z$ in the domain. We leverage the same procedure as for our MeGA-MP (cf. Eq. 8) and compute $\delta t(z)$ as the unique solution of $\int_0^{\delta t(z)} \nu(s) ds = z$. Since $\bar{\nu}(t)$ is strictly increasing the solution is unique and well-defined. According to Theorem C.11, the coordinate $(t + \delta t(z), z)$ lies on the characteristic curve that intersects the boundary at $(t, 0)$. We can shift the boundary value Gaussian accordingly and obtain $c_{bvp}(t, z) = c(t + \delta t(z), 0)$. The solution to the initial value problem with boundary condition is then

$$c(t, z) = c_{ivp}(t, z) + c_{bvp}(t, z)$$

In practice, we compute the respective values numerically at each discrete spatial and temporal grid point.

Next to the analytical solution, the numerical and analytical solvers we compare ourselves to are:

- **Semi-Lagrangian scheme**: We use a custom semi-Lagrangian solver. For every grid point $z \in \Omega$ and iteration step $n+1$, the method computes the departure point $z'$ of a particle arriving at $z$ by tracing backward along the velocity field $\nu(t, z)$. Since the departure point $z'$ is a continuous coordinate, we interpolate spatially to compute the new solution $c(t_{n+1}, z)$. If we assume a spatially constant and temporally piecewise constant flow field, the departure point is simply $z' = z - dt \cdot \nu(t_n, z)$. Otherwise, it is approximated by integrating the characteristic ODE $\frac{dz}{dt} = -\nu(t, z)$ via the RK4 method. The result is then $c(t_{n+1}, z) = \mathcal{I}[c(t_n, \cdot)](z')$, where $\mathcal{I}$ is the spatial interpolation operator like the linear interpolation in Eq. (16).
- **Runge-Kutta-4** is the classical order-four Runge-Kutta (RK4) numerical method for solving PDEs (and ODEs) of the kind

$$\frac{\partial c(t, z)}{\partial t} = f(t, z, c)$$

with the update rule

$$c(t_{n+1}, z) = c(t_n, z) + \frac{dt}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

where $n \in \mathbb{N}$ is the iteration step, $dt$ is the step-size and $k_1 = f(t_n, z, c)$, $k_2 = f(t_n + \frac{dt}{2}, c(t_n, z) + \frac{dt}{2}k_1)$, $k_3 = f(t_n + \frac{dt}{2}, c(t_n, z) + \frac{dt}{2}k_2)$ and $k_4 = f(t_n + dt, c(t_n, z) + dtk_3)$. We apply the RK4 within a method-of-lines setup. The right-hand side function $f(t, z, c)$ approximates the negative spatial derivative of the flux via a fifth-order Weighted Essentially Non-Oscillatory (WENO5) scheme (Jiang & Shu, 1996) using Lax-Friedrichs flux-splitting for numerical stability. Specifically, left- and right-moving flux components are reconstructed separately at cell interfaces, resulting in the spatial derivative approximation at cell grid-points $i$: $f(t, z_i, c) \approx -\frac{\hat{F}_{i+1/2} - \hat{F}_{i-1/2}}{dx}$ where $\hat{F}_{i\pm 1/2}$ represent the combined numerical interface fluxes.

**Baselines and Results** In Figure 17, the analytical solution of the boundary value problem, which serves as ground truth, and the MeGA-MP solution are visualized at different discretizations and on the whole resulting grid. Moreover, in Figure 18, the analytical solution of the initial value problem with boundary values, which serves as ground truth, and the MeGA-MP solution are visualized at different discretizations and on the whole resulting grid. Additionally, in Figure 19, we visualize the solutions of the initial value problem with boundary conditions (34) for different solvers and different temporal and spatial discretizations at the last step $t = 100$.

As an extension of Table 3, in Table 5, we compare our method to the numerical solvers at different discretizations. We also report the runtime of each method in Table 6. It can be observed that a larger number of edges (smaller $dx$) results in longer runtime, which is expected. Note that these values are only meant to convey a rough idea of time complexity since there may be several ways to improve the performance of the baselines. We ran all methods on the CPU.

# E Proofs

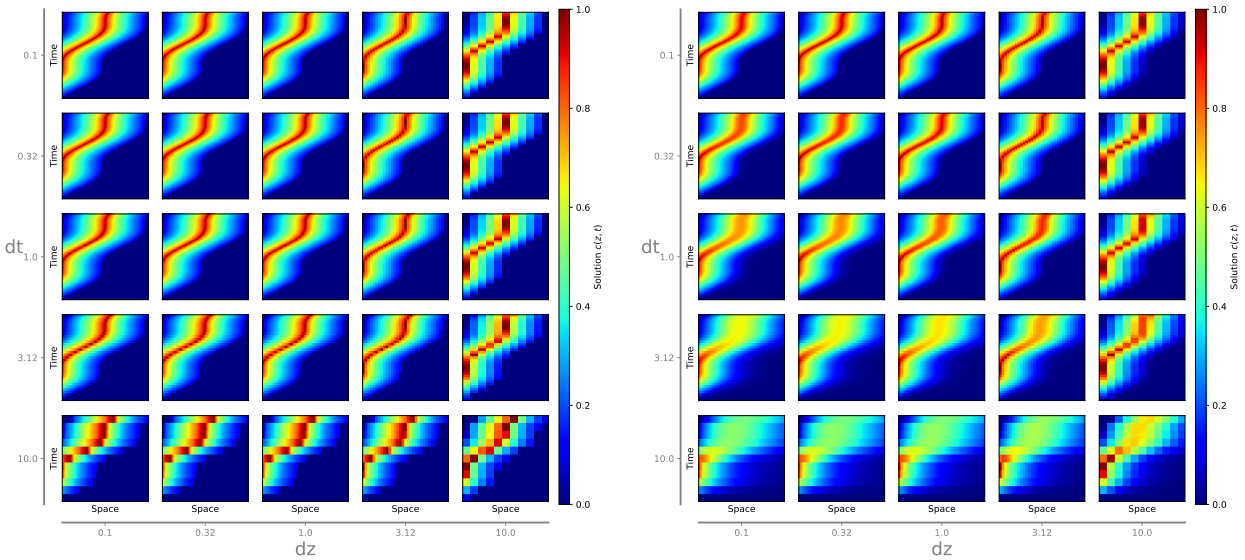
## E.1 Proof of Lemma 3.2 and C.2

**Lemma** (Lemma 3.2 and C.2)**.** *Let $e_{uv} \in E$ hold. If the function $c_{uv}$ obeys Equation (2), for each $t_0 \in \mathbb{R}$ and $z_0 \in \mathbb{R}$, the curve*

$$\begin{aligned} \gamma_{t_0,z_0} : \; T_{t_0,z_0} &\longrightarrow \mathbb{R}_{\geq 0} \\ t &\longmapsto c_{uv}(t, z_0 + \textstyle\int_{t_0}^{t} \nu_{uv}(s)\; ds) \end{aligned}$$

*is constant on*

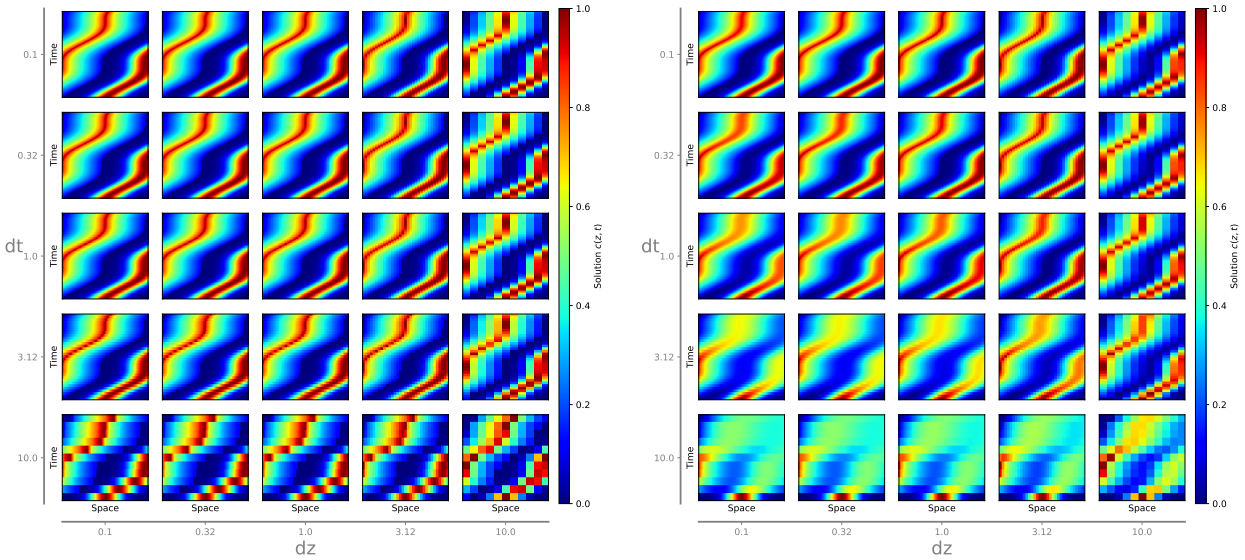
$$T_{t_0,z_0} = \Big\{ t \in \mathbb{R} \mid 0 < z_0 + \int_{t_0}^{t'} \nu_{uv}(s)\; ds < l_{uv} \;\; \forall t' \in \big[\min\{t_0, t\}, \max\{t_0, t\}\big] \Big\}.$$




(a) Analytical solution.

(b) MeGA-MP solution.

Figure 17: Solutions of the **boundary value problem** (33) for different temporal and spatial discretizations and on the whole resulting grid




(a) Analytical solution.

(b) MeGA-MP solution.

Figure 18: Solutions of **initial value problem** with boundary condition (34) for different temporal and spatial discretizations and on the whole resulting grid.

*Proof.* Let $c_{uv} : \mathbb{R} \times \Omega_{uv} \to \mathbb{R}_{\geq 0}$ obey Equation (2) (note that Equation (2) is only defined for $z \in \Omega_{uv} = (0, l_{uv})$). We can consider its space coordinate $z$ as a function $z : \mathbb{R} \to \Omega_{uv}$ parametrized by time. Consequently, we can use the method of characteristics (cf. Habermann (2013)) to derive the following system of ODEs from Equation (2) for the composed, now only time-dependent function $c_{uv} : \mathbb{R} \to \mathbb{R}_{\geq 0}, c_{uv}(t) := c_{uv}(t, z(t))$:

$$\frac{dc_{uv}(t)}{dt} = 0, \quad \frac{dz(t)}{dt} = \nu_{uv}(t), \quad \frac{dt}{dt} = 1.$$

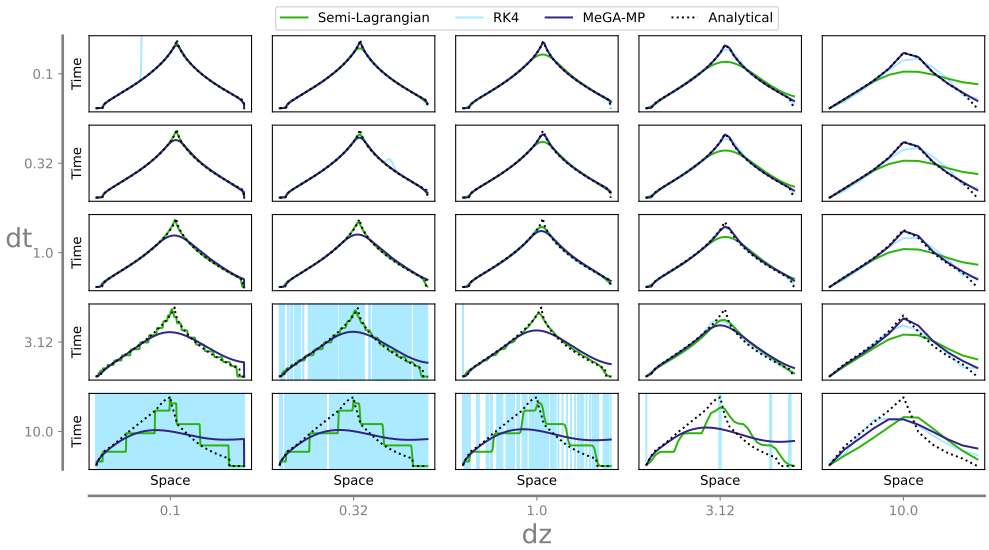



Figure 19: Solutions of the initial value problem with boundary condition (33) for different solvers and different temporal and spatial discretizations at the last step $t = 100$.

Table 5: MAE between analytical ground truth and different solver solutions over all temporal and spatial grid points. The entry "–" indicates divergence of the method.

| $dt$ | $dx$ | semi- Lagr. | RK4 | MeGA-MP |
|---|---|---|---|---|
| 0.1 | 0.10 | **0.0016**$\pm$0.0050 | – | 0.0024$\pm$0.0081 |
| | 0.32 | 0.0046$\pm$0.0114 | **0.0013**$\pm$0.0043 | **0.0013**$\pm$0.0058 |
| | 1.00 | 0.0108$\pm$0.0230 | 0.0035$\pm$0.0073 | **0.0012**$\pm$0.0068 |
| | 3.12 | 0.0239$\pm$0.0422 | 0.0105$\pm$0.0152 | **0.0023**$\pm$0.0113 |
| | 10.00 | 0.0562$\pm$0.0764 | 0.0288$\pm$0.0415 | **0.0057**$\pm$0.0182 |
| 0.32 | 0.10 | 0.0027$\pm$0.0060 | – | 0.0071$\pm$0.0162 |
| | 0.32 | **0.0038**$\pm$0.0075 | 0.0108$\pm$0.0325 | 0.0049$\pm$0.0123 |
| | 1.00 | 0.0095$\pm$0.0184 | 0.0042$\pm$0.0082 | **0.0031**$\pm$0.0092 |
| | 3.12 | 0.0227$\pm$0.0400 | 0.0117$\pm$0.0160 | **0.0036**$\pm$0.0126 |
| | 10.00 | 0.0557$\pm$0.0754 | 0.0304$\pm$0.0418 | **0.0071**$\pm$0.0189 |
| 1.0 | 0.10 | 0.0080$\pm$0.0144 | – | 0.0157$\pm$0.0298 |
| | 0.32 | 0.0080$\pm$0.0129 | – | 0.0143$\pm$0.0277 |
| | 1.00 | 0.0097$\pm$0.0137 | **0.0088**$\pm$0.0155 | 0.0103$\pm$0.0208 |
| | 3.12 | 0.0206$\pm$0.0324 | 0.0100$\pm$0.0155 | **0.0081**$\pm$0.0165 |
| | 10.00 | 0.0545$\pm$0.0720 | 0.0291$\pm$0.0395 | **0.0115**$\pm$0.0222 |
| 3.12 | 0.10 | 0.0250$\pm$0.0364 | – | 0.0384$\pm$0.0535 |
| | 0.32 | **0.0243**$\pm$0.0350 | – | 0.0375$\pm$0.0524 |
| | 1.00 | **0.0231**$\pm$0.0315 | – | 0.0345$\pm$0.0491 |
| | 3.12 | 0.0254$\pm$0.0321 | **0.0231**$\pm$0.0333 | 0.0242$\pm$0.0370 |
| | 10.00 | 0.0504$\pm$0.0639 | 0.0299$\pm$0.0406 | **0.0238**$\pm$0.0314 |
| 10.0 | 0.10 | **0.0763**$\pm$0.0964 | – | 0.0939$\pm$0.0981 |
| | 0.32 | **0.0752**$\pm$0.0954 | – | 0.0933$\pm$0.0977 |
| | 1.00 | **0.0728**$\pm$0.0923 | – | 0.0915$\pm$0.0952 |
| | 3.12 | **0.0675**$\pm$0.0852 | – | 0.0843$\pm$0.0904 |
| | 10.00 | **0.0671**$\pm$0.0842 | 0.0728$\pm$0.0936 | 0.0761$\pm$0.0763 |

By the fundamental theorem of calculus, we observe that for each $t_0 \in \mathbb{R}$ and $z_0 \in \mathbb{R}$, the function

$$z : \mathbb{R} \longrightarrow \mathbb{R},\ t \longmapsto z(t) := z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds$$

Table 6: Time in seconds that it takes each method to solve the BVP. The time it takes for the preprocessing to convert flows into transport times (s. Algorithm 1) is included in these values. The entry "–" indicates divergence of the method.

| $dt$ | $dx$ | semi- Lagr. | RK4 | MeGA-MP | Analytical |
|---|---|---|---|---|---|
| 0.1 | 0.10 | **58.1295** | – | 63.1235 | 0.2867 |
| | 0.32 | 16.9899 | **1.0589** | **5.9495** | 0.0792 |
| | 1.00 | 5.5392 | **0.8999** | **1.3102** | 0.0416 |
| | 3.12 | 1.7895 | 0.8881 | **0.5884** | 0.0097 |
| | 10.00 | 0.6871 | 0.8863 | **0.1444** | 0.0039 |
| 0.32 | 0.10 | 18.0842 | – | **13.9027** | 0.0398 |
| | 0.32 | 5.4553 | **0.3194** | 1.7582 | 0.0130 |
| | 1.00 | 1.6860 | **0.2795** | 0.5185 | 0.0049 |
| | 3.12 | 0.6071 | 0.2794 | **0.2051** | 0.0019 |
| | 10.00 | **0.2222** | 0.2766 | 0.3668 | 0.0013 |
| 1.0 | 0.10 | 6.0633 | – | **6.0126** | 0.0109 |
| | 0.32 | 1.7784 | – | **1.1763** | 0.0029 |
| | 1.00 | 0.5320 | **0.0893** | 0.2651 | 0.0012 |
| | 3.12 | 0.1810 | 0.0883 | **0.0628** | 0.0007 |
| | 10.00 | 0.0699 | 0.0871 | **0.0218** | 0.0005 |
| 3.12 | 0.10 | **1.7991** | – | 3.6538 | 0.0024 |
| | 0.32 | **0.5470** | – | 0.7568 | 0.0012 |
| | 1.00 | 0.1691 | – | **0.1539** | 0.0007 |
| | 3.12 | 0.0582 | **0.0283** | 0.0470 | 0.0006 |
| | 10.00 | 0.0223 | 0.0281 | **0.0191** | 0.0004 |
| 10.0 | 0.10 | **0.5620** | – | 2.3683 | 0.0012 |
| | 0.32 | **0.1711** | – | 0.4693 | 0.0008 |
| | 1.00 | **0.0570** | – | 0.1159 | 0.0005 |
| | 3.12 | **0.0189** | – | 0.0400 | 0.0005 |
| | 10.00 | **0.0075** | 0.0088 | 0.0172 | 0.0005 |

satisfies the second ODE. Therefore, by the first ODE, we can conclude that the function $c_{uv}$ is constant along characteristic curves $(t, z(t)) = (t, z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds)$ along the time $\mathbb{R}$ as well as along the space interval $\Omega_{uv}$, i.e., *within* the edge $e_{uv}$ of length $l_{uv} > 0$. Therefore, taking into account the domain and co-domain of $c_{uv}$, $\nu_{uv}$ and $z$ as a function, $t_0, t \in \mathbb{R}$ and $z(t^{'}) \in \Omega_{uv} = (0, l_{uv})$ need to hold for all $t^{'} \in [\min\{t_0, t\}, \max\{t_0, t\}]$.[17]

In summary, the curve $\gamma_{z_0} : T_{t_0,z_0} \to \mathbb{R}_{\geq 0}, t \mapsto c_{uv}(t, z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds)$ is constant on

$$T_{t_0,z_0} = \Big\{ t \in \mathbb{R} \mid 0 < z_0 + \int_{t_0}^{t^{'}} \nu_{uv}(s)\ ds < l_{uv}\ \forall t^{'} \in \big[\min\{t_0, t\}, \max\{t_0, t\}\big] \Big\}.$$

□

### E.2 Proof of Lemma C.4

**Lemma** (Lemma C.4)**.** *Let $e_{uv} \in E$, $t_0 \in \mathbb{R}$, $z_0 \in \mathbb{R}$ hold and let $T_{t_0,z_0}$ be defined as in Lemma 3.2.*

1. *If $z_0 \notin \Omega_{uv}$ holds, then $T_{t_0,z_0} = \emptyset$ holds.*
2. *If $z_0 \in \Omega_{uv}$ holds, then $T_{t_0,z_0} \neq \emptyset$ holds. Even more, $t_0 \in T_{t_0,z_0}$ holds.*
3. *If $z_0 \in \Omega_{uv}$ holds, there exist $t_l \in [-\infty, t_0)$ and $t_r \in (t_0, \infty]$ such that*

[17] Especially to mention, the requirement $z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds \in \Omega_{uv}$ does not suffices, cf. Remark C.3.

*(a) if $t_l \neq -\infty$ holds, $z_0 + \int_{t_0}^{t_l} \nu_{uv}(s)\ ds \in \{0, l_{uv}\}$ holds,*
*(b) if $t_r \neq +\infty$ holds, $z_0 + \int_{t_0}^{t_r} \nu_{uv}(s)\ ds \in \{0, l_{uv}\}$ holds and*
*(c) $T_{t_0,z_0} = (t_l, t_r) = (t_0 - (t_0 - t_l), t_0 + (t_r - t_0))$ holds.*

*Equivalently, there exist $\delta t_l, \delta t_r \in (0, \infty]$ such that $T_{t_0,z_0} = (t_0 - \delta t_l, t_0 + \delta t_r)$ holds.*

*Proof.* Similarly to Remark C.3, we define the function

$$z : \mathbb{R} \longrightarrow \mathbb{R},\ t \longmapsto z(t) := z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds.$$

By the fundamental theorem of calculus, the function $z$ is differentiable and therefore, continuous. Moreover, using the function $z$, we can re-write the set $T_{t_0,z_0}$ as

$$T_{t_0,z_0} = \Big\{ t \in \mathbb{R} \mid z(t^{'}) \in \Omega_{uv} = (0, l_{uv})\ \forall t^{'} \in \big[\min\{t_0, t\}, \max\{t_0, t\}\big] \Big\}.$$

*1.:* If $z_0 \notin \Omega_{uv}$ holds, let us assume that $T_{t_0,z_0} \neq \emptyset$ holds. By definition of $T_{t_0,z_0}$, there exists a $t \in \mathbb{R}$, such that $z(t^{'}) \in \Omega_{uv}$ holds for all $t^{'} \in [\min\{t_0, t\}, \max\{t_0, t\}]$. However, for $t^{'} = t_0 \in [\min\{t_0, t\}, \max\{t_0, t\}]$, we obtain $z(t^{'}) = z(t_0) = z_0 + \int_{t_0}^{t_0} \nu_{uv}(s)\ ds = z_0 \notin \Omega_{uv}$ – a contradiction.

*2.:* If $z_0 \in \Omega_{uv}$ holds, we obtain $z(t_0) = z_0 + \int_{t_0}^{t_0} \nu_{uv}(s)\ ds = z_0 \in \Omega_{uv}$ and therefore, trivially, $t_0 \in T_{t_0,z_0}$ and $T_{t_0,z_0} \neq \emptyset$ holds.

*3.:* We define the sets

$$\begin{aligned} A_{\mathrm{l}} :=& \{t^{'} \in (-\infty, t_0] \mid z(t^{'}) \notin \Omega_{uv}\} \text{ and} \\ A_{\mathrm{r}} :=& \{t^{'} \in [t_0, +\infty) \mid z(t^{'}) \notin \Omega_{uv}\}. \end{aligned}$$

Based on these sets, we choose

$$t_{\mathrm{l}} := \begin{cases} \max A_{\mathrm{l}} & \text{if } A_{\mathrm{l}} \neq \emptyset \\ -\infty & \text{if } A_{\mathrm{l}} = \emptyset \end{cases} \quad \text{and} \quad t_{\mathrm{r}} := \begin{cases} \min A_{\mathrm{r}} & \text{if } A_{\mathrm{r}} \neq \emptyset \\ +\infty & \text{if } A_{\mathrm{r}} = \emptyset \end{cases}.$$

If $z_0 \in \Omega_{uv}$ holds, as seen above, $z(t_0) = z_0 \in \Omega_{uv}$ holds. Therefore, we observe that $t_0 \notin A_{\mathrm{l}}$ and $t_0 \notin A_{\mathrm{r}}$, or equivalently, $A_{\mathrm{l}} \subset (-\infty, t_0)$ and $A_{\mathrm{r}} \subset (t_0, +\infty)$ hold. Consequently, our choices of $t_{\mathrm{l}}$ and $t_{\mathrm{r}}$ indeed satisfy $t_{\mathrm{l}} \in [-\infty, t_0)$ and $t_{\mathrm{r}} \in (t_0, \infty]$ and even more, $t_0 \in (t_{\mathrm{l}}, t_{\mathrm{r}})$ holds.

We now want to show that for these choices, indeed, *5(a) - 5(c)* hold.

*3(a), 3(b): If $t_l \neq -\infty$ holds, $z_0 + \int_{t_0}^{t_l} \nu_{uv}(s)\ ds \in \{0, l_{uv}\}$ holds and if $t_r \neq +\infty$ holds, $z_0 + \int_{t_0}^{t_r} \nu_{uv}(s)\ ds \in \{0, l_{uv}\}$ holds.*

The claims are equivalent to show that if $A_{\mathrm{l}} \neq \emptyset$ holds, $z(t_{\mathrm{l}}) \in \{0, l_{uv}\}$ holds and if $A_{\mathrm{r}} \neq \emptyset$ holds, $z(t_{\mathrm{r}}) \in \{0, l_{uv}\}$ holds.

If $A_{\mathrm{l}} \neq \emptyset$ holds, let us assume that $z(t_{\mathrm{l}}) \notin \{0, l_{uv}\}$ holds. Since $t_{\mathrm{l}} = \max A_{\mathrm{l}} \in A_{\mathrm{l}}$, $z(t_{\mathrm{l}}) > l_{uv}$ or $z(t_{\mathrm{l}}) < 0$ must hold. Without loss of generality (W.l.o.g.), let us assume that $z(t_{\mathrm{l}}) < 0$ holds. Since $z$ is continuous and $z(t_0) = z_0 \in \Omega_{uv}$, i.e., $0 < z(t_0)$ holds, according to the intermediate value theorem, there exists a $t^{'} \in (t_{\mathrm{l}}, t_0)$ such that $z(t^{'}) = 0$ holds. Therefore, we can conclude that $\max A_{\mathrm{l}} = t_{\mathrm{l}} < t^{'} \in A_{\mathrm{l}}$ holds – a contradiction to the choice of $t_{\mathrm{l}} = \max A_{\mathrm{l}}$.

If $A_{\mathrm{r}} \neq \emptyset$ holds, let us assume $z(t_{\mathrm{r}}) \notin \{0, l_{uv}\}$ holds. Analogous arguments lead to a contradiction.

*3(c), step 1: $T_{t_0,z_0} \subset (t_l, t_r)$ holds.*

Let $t \in T_{t_0,z_0} \subset \mathbb{R}$ and let us assume that $t \notin (t_\mathrm{l}, t_\mathrm{r})$, or equivalently, $t \leq t_\mathrm{l} < t_0$ or $t \geq t_\mathrm{r} > t_0$ holds.
*Case 1:* If $t \leq t_0$ and $A_\mathrm{l} \neq \emptyset$ holds, we observe that $t \leq t_\mathrm{l} = \max A_\mathrm{l} < t_0$ must hold. By definition of $T_{t_0,z_0}$, $z(t^{'}) \in \Omega_{uv}$ holds for all $t^{'} \in [\min\{t_0, t\}, \max\{t_0, t\}] = [t, t_0]$. However, by definition of $A_\mathrm{l}$, for $t^{'} = t_\mathrm{l} \in [t, t_0) \subset [t, t_0]$, $z(t^{'}) = z(t_\mathrm{l}) \notin \Omega_{uv}$ holds – a contradiction to the first conclusion.
*Case 2:* If $t \leq t_0$ and $A_\mathrm{l} = \emptyset$ holds, we observe that $t \leq t_\mathrm{l} = -\infty$ must hold – a contradiction.
*Case 3:* If $t \geq t_0$ and $A_\mathrm{l} \neq \emptyset$ holds, we observe that $t \geq t_\mathrm{r} = \min A_\mathrm{r} > t_0$ must hold. By definition of $T_{t_0,z_0}$, $z(t^{'}) \in \Omega_{uv}$ holds for all $t^{'} \in [\min\{t_0, t\}, \max\{t_0, t\}] = [t_0, t]$. However, by definition of $A_\mathrm{r}$, for $t^{'} = t_\mathrm{r} \in (t_0, t] \subset [t, t_0]$, $z(t^{'}) = z(t_\mathrm{r}) \notin \Omega_{uv}$ holds – a contradiction to the first conclusion.
*Case 4:* If $t \geq t_0$ and $A_\mathrm{l} = \emptyset$ holds, we observe that $t \geq t_\mathrm{r} = +\infty$ must hold – a contradiction.

*3(c), step 2: $(t_l, t_r) \subset T_{t_0,z_0}$ holds.*

Let $t \in (t_\mathrm{l}, t_\mathrm{r}) \subset \mathbb{R}$ and let us assume that $t \notin T_{t_0,z_0}$ holds.
*Case 1:* If $t \leq t_0$ and $A_\mathrm{l} \neq \emptyset$ holds, by definition of $T_{t_0,z_0}$, there exists a $t^{'} \in [\min\{t_0, t\}, \max\{t_0, t\}] = [t, t_0] \subset (t_\mathrm{l}, t_0] = (\max A_\mathrm{l}, t_0]$ such that $z(t^{'}) \notin \Omega_{uv}$ holds. Therefore, we can conclude that $\max A_\mathrm{l} = t_\mathrm{l} < t^{'} \in A_\mathrm{l}$ holds – a contradiction to the choice of $t_\mathrm{l} = \max A_\mathrm{l}$.
*Case 2:* If $t \leq t_0$ and $A_\mathrm{l} \neq \emptyset$ holds, by definition of $T_{t_0,z_0}$, there exists a $t^{'} \in [\min\{t_0, t\}, \max\{t_0, t\}] = [t, t_0] \subset (t_\mathrm{l}, t_0] = (-\infty, t_0]$ such that $z(t^{'}) \notin \Omega_{uv}$ holds. Therefore, we can conclude that $t^{'} \in A_\mathrm{l}$ holds – a contradiction to $A_\mathrm{l} \neq \emptyset$.
*Case 3:* If $t \geq t_0$ and $A_\mathrm{r} \neq \emptyset$ holds, by definition of $T_{t_0,z_0}$, there exists a $t^{'} \in [\min\{t_0, t\}, \max\{t_0, t\}] = [t_0, t] \subset [t_0, t_\mathrm{r}) = [t_0, \min A_\mathrm{r})$ such that $z(t^{'}) \notin \Omega_{uv}$ holds. Therefore, we can conclude that $\min A_\mathrm{r} = t_\mathrm{r} > t^{'} \in A_\mathrm{r}$ holds – a contradiction to the choice of $t_\mathrm{r} = \min A_\mathrm{l}$.
*Case 4:* If $t \geq t_0$ and $A_\mathrm{r} \neq \emptyset$ holds, by definition of $T_{t_0,z_0}$, there exists a $t^{'} \in [\min\{t_0, t\}, \max\{t_0, t\}] = [t_0, t] \subset [t_0, t_\mathrm{r}) = [t_0, +\infty)$ such that $z(t^{'}) \notin \Omega_{uv}$ holds. Therefore, we can conclude that $t^{'} \in A_\mathrm{r}$ holds – a contradiction to $A_\mathrm{r} \neq \emptyset$. □

### E.3 Proof of Theorem C.5

**Theorem** (Theorem C.5)**.** *Let $e_{uv} \in E$ hold. If the function $c_{uv}$ obeys Equation* (2)*, for each $t_0 \in \mathbb{R}$ and $z_0 \in \Omega_{uv}$,*

$$c_{uv}\left(t, z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds\right) = c_{uv}\left(t, z_0 + \overline{\nu}_{uv}(t_0, t)\ (t - t_0)\right) = c_{uv}(t_0, z_0)$$

*holds for all $t \in T_{t_0,z_0}$ with $T_{t_0,z_0}$ as defined in Lemma 3.2 and for the mean velocity*

$$\overline{\nu}_{uv}(t_0, t) := \begin{cases} \fint_{t_0}^{t} \nu_{uv}(s)\ ds & \text{if } t \neq t_0 \\ 0 & \text{if } t = t_0 \end{cases} = \begin{cases} \frac{1}{t-t_0} \int_{t_0}^{t} \nu_{uv}(s)\ ds & \text{if } t \neq t_0 \\ 0 & \text{if } t = t_0 \end{cases}.$$

*Proof.* For each $t_0 \in \mathbb{R}$ and $z_0 \in \Omega_{uv} \subset \mathbb{R}$, by Lemma 3.2, $\gamma_{t_0,z_0}(t) = c_{uv}(t, z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds) = c$ holds for all $t \in T_{t_0,z_0}$ and a constant $c \in \mathbb{R}_{\geq 0}$. Thus, it suffices to show that $c = c_{uv}(t_0, z_0)$ holds.

Let us assume that $c \neq c_{uv}(t_0, z_0)$ holds. Since $z_0 \in \Omega_{uv}$, by Lemma C.4, $t_0 \in T_{t_0,z_0}$ holds. However, for $t = t_0 \in T_{t_0,z_0}$, we obtain $c = \gamma_{t_0,z_0}(t) = c_{uv}(t, z_0 + \int_{t_0}^{t_0} \nu_{uv}(s)\ ds) = c_{uv}(t_0, z_0)$ – a contradiction. Therefore, $c_{uv}(t, z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds) = c_{uv}(t_0, z_0)$ holds. Finally, by definition of $\overline{\nu}_{uv}(t_0, t)$, for $t \neq t_0$, we obtain

$$\overline{\nu}_{uv}(t_0, t) \cdot (t - t_0) = \frac{t - t_0}{t - t_0} \int_{t_0}^{t} \nu_{uv}(s)\ ds = \int_{t_0}^{t} \nu_{uv}(s)\ ds,$$

allowing to conclude that $c_{uv}(t, z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds) = c_{uv}(t, z_0 + \overline{\nu}_{uv}(t_0, t)\ (t - t_0))$ holds. □

### E.4 Proof of Lemma C.7

**Lemma** (Lemma C.7)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. If there exists a pass-through time $\delta t_{uv} \in \mathbb{R}$ w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$, the following properties hold:*

1. *$\delta t_{uv} \in \mathbb{R}_{\neq 0}$ and thus, $t - \delta t_{uv} \neq t$ holds.*
2. *There exists no other pass-through time of the same sign (i.e., there exists either exactly one or no positive pass-through time, and there exists exactly one or no negative pass-through time).*
3. *The mean velocity*

$$\overline{\nu}_{uv}(t) := \overline{\nu}_{uv}(t - \delta t_{uv}, t) = \fint_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds = \frac{1}{\delta t_{uv}} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds$$

*during the time interval $[\min\{t - \delta t_{uv}, t\}, \max\{t - \delta t_{uv}, t\}]$ as defined in Theorem C.5 is equal to $\overline{\nu}_{uv}(t) = \frac{l_{uv}}{\delta t_{uv}}$ and thus satisfies*

$$\overline{\nu}_{uv}(t) > 0 \Longleftrightarrow \delta t_{uv} > 0 \text{ and}$$
$$\overline{\nu}_{uv}(t) < 0 \Longleftrightarrow \delta t_{uv} < 0.$$

4. *The flows $q_{uv}(t - \delta t_{uv})$ and $q_{uv}(t)$ linked to the velocity $\nu_{uv}(\cdot) = \frac{q_{vu(\cdot)}}{\alpha_{vu}}$ satisfy*

$$\begin{cases} q_{uv}(t - \delta t_{uv}), q_{uv}(t) \geq 0 & \text{if } \delta t_{uv} > 0 \\ q_{uv}(t - \delta t_{uv}), q_{uv}(t) \leq 0 & \text{if } \delta t_{uv} < 0 \end{cases}.$$

*Proof. 1.:* Let us assume that $\delta t_{uv} = 0$ holds. Consequently, (1) by basic results from analysis, (2) condition (19) and (3) definition of the pipe length $l_{uv} > 0$,

$$0 \overset{(1)}{=} \int_{t-0}^{t} \nu_{uv}(s)\ ds \overset{(1)}{=} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(2)}{=} l_{uv} \overset{(3)}{>} 0$$

holds – a contradiction.

*2.:* Let us assume that there exist two different pass-through times $\delta t_{1uv}, \delta t_{2uv} \in \mathbb{R}_{\neq 0}$ with the same sign. If $\delta t_{1uv}, \delta t_{2uv} > 0$ holds, without loss of generality (w.l.o.g.), let $0 < \delta t_{1uv} < \delta t_{2uv}$ hold. Consequently, for $t^{'} = t - \delta t_{1uv} \in (t - \delta t_{2uv}, t) = (\min\{t - \delta t_{2uv}, t\}, \max\{t - \delta t_{2uv}, t\})$, by (1) by basic results from analysis and (2) condition (19), we obtain

$$\int_{t-\delta t_{2uv}}^{t^{'}} \nu_{uv}(s)\ ds \overset{(1)}{=} \int_{t-\delta t_{2uv}}^{t} \nu_{uv}(s)\ ds - \int_{t-\delta t_{1uv}}^{t} \nu_{uv}(s)\ ds \overset{(2)}{=} l_{uv} - l_{uv} \overset{(1)}{=} 0$$

– a contradiction to condition (20).

If $\delta t_{1uv}, \delta t_{2uv} < 0$ holds, w.l.o.g., let $\delta t_{2uv} < \delta t_{1uv} < 0$ hold. Consequently, for $t^{'} = t - \delta t_{1uv} \in (t, t - \delta t_{2uv}) = (\min\{t - \delta t_{2uv}, t\}, \max\{t - \delta t_{2uv}, t\})$, we can use the same calculation to again obtain a contradiction.

*3.:* By (1) definition of the mean velocity $\overline{\nu}_{uv}(t - \delta t_{uv}, t)$ as defined in Theorem C.5 and (2) condition (19), we indeed obtain

$$\overline{\nu}_{uv}(t) := \overline{\nu}_{uv}(t - \delta t_{uv}, t) \overset{(1)}{=} \fint_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(1)}{=} \frac{1}{\delta t_{uv}} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(2)}{=} \frac{l_{uv}}{\delta t_{uv}}.$$

The second claim then follows by the fact that by definition of the pipe length, $l_{uv} > 0$ holds.

*4.:* If $\delta t_{uv} > 0$ holds, let us assume that $q_{uv}(t - \delta t_{uv}) < 0$, and consequently, $\nu_{uv}(t - \delta t_{uv}) < 0$ holds. Since $\nu_{uv} \in C^0(\mathbb{R}, \mathbb{R})$ is continuous, there exists a $t^{'} \in (t - \delta t_{uv}, t) = (\min\{t - \delta t_{uv}, t\}, \max\{t - \delta t_{uv}, t\})$ such that $\nu_{uv}(s) < 0$ holds for all $s \in (t - \delta t_{uv}, t^{'})$. Consequently, by (1) basic results from analysis, we obtain

$$\int_{t-\delta t_{uv}}^{t^{'}} \underbrace{\nu_{uv}(s)}_{<0} \; ds \overset{(1)}{<} 0$$

– a contradiction to condition (20).

Similarly, if $\delta t_{uv} > 0$ holds, let us assume that $q_{uv}(t) < 0$, and consequently, $\nu_{uv}(t) < 0$ holds. Since $\nu_{uv} \in C^0(\mathbb{R}, \mathbb{R})$ is continuous, there exists a $t^{'} \in (t - \delta t_{uv}, t) = (\min\{t - \delta t_{uv}, t\}, \max\{t - \delta t_{uv}, t\})$ such that $\nu_{uv}(s) < 0$ holds for all $s \in (t^{'}, t)$. Consequently, by (1) basic results from analysis and (2) condition (19), we obtain

$$\int_{t-\delta t_{uv}}^{t^{'}} \nu_{uv}(s) \; ds \overset{(1)}{=} \underbrace{\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s) \; ds}_{\overset{(2)}{=} l_{uv}} - \underbrace{\underbrace{\int_{t^{'}}^{t} \underbrace{\nu_{uv}(s)}_{<0} \; ds}_{\overset{(1)}{<} 0}}_{>0} > l_{uv}$$

– again a contradiction to condition (20).

In contrast, if $\delta t_{uv} < 0$ holds, let us assume that $q_{uv}(t - \delta t_{uv}) > 0$, and consequently, $\nu_{uv}(t - \delta t_{uv}) > 0$ holds. Since $\nu_{uv} \in C^0(\mathbb{R}, \mathbb{R})$ is continuous, there exists a $t^{'} \in (t, t - \delta t_{uv}) = (\min\{t - \delta t_{uv}, t\}, \max\{t - \delta t_{uv}, t\})$ such that $\nu_{uv}(s) > 0$ holds for all $s \in (t^{'}, t - \delta t_{uv})$. Consequently, by (1) basic results from analysis, we obtain

$$\int_{t-\delta t_{uv}}^{t^{'}} \nu_{uv}(s) \; ds \overset{(1)}{=} - \underbrace{\int_{t^{'}}^{t-\delta t_{uv}} \underbrace{\nu_{uv}(s)}_{>0} \; ds}_{\overset{(1)}{>} 0} < 0$$

– again a contradiction to condition (20).

Similarly, if $\delta t_{uv} < 0$ holds, let us assume that $q_{uv}(t) > 0$, and consequently, $\nu_{uv}(t) > 0$ holds. Since $\nu_{uv} \in C^0(\mathbb{R}, \mathbb{R})$ is continuous, there exists a $t^{'} \in (t, t - \delta t_{uv}) = (\min\{t - \delta t_{uv}, t\}, \max\{t - \delta t_{uv}, t\})$ such that $\nu_{uv}(s) > 0$ holds for all $s \in (t, t^{'})$. Consequently, by (1) basic results from analysis and (2) condition (19), we obtain

$$\begin{aligned}\int_{t-\delta t_{uv}}^{t^{'}} \nu_{uv}(s) \; ds &\overset{(1)}{=} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s) \; ds - \int_{t^{'}}^{t} \nu_{uv}(s) \; ds \\ &\overset{(1)}{=} \underbrace{\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s) \; ds}_{\overset{(2)}{=} l_{uv}} + \underbrace{\underbrace{\int_{t}^{t^{'}} \underbrace{\nu_{uv}(s)}_{>0} \; ds}_{\overset{(1)}{>} 0}}_{>0} > l_{uv}\end{aligned}$$

– again a contradiction to condition (20).

*Remark* E.1. Note that if one of the flow rates discussed above are zero, that they can only be zero in this point of time, but not on a whole sub-interval of the time interval $[\min\{t - \delta t_{uv}, t\}, \max\{t - \delta t_{uv}, t\}]$.

More precisely, if $\delta t_{uv} > 0$ holds, let us assume that there exists an $\epsilon > 0$ such that for all $s \in [t - \delta t_{uv}, t - \delta t_{uv} + \epsilon)$, $q_{uv}(s) = 0$, and consequently, $\nu_{uv}(s) = 0$, holds. Thus, for $t^{'} = \min\{t - \delta t_{uv} + \epsilon, t\} \in (t - \delta t_{uv}, t) =$

$(\min\{t-\delta t_{uv}, t\}, \max\{t-\delta t_{uv}, t\})$, by (1) basic results from analysis, we obtain

$$\int_{t-\delta t_{uv}}^{t'} \underbrace{\nu_{uv}(s)}_{=0} \; ds \overset{(1)}{=} 0$$

– a contradiction to condition (20). In combination with the findings above, therefore, $q_{uv}(t-\delta t_{uv}) \geq 0$ needs to hold, and if the (unlikely) case $q_{uv}(t-\delta t_{uv}) = 0$ indeed applies, we observe that for all $\epsilon > 0$ there exists an $s \in [t-\delta t_{uv}, t-\delta t_{uv}+\epsilon)$ such that $q_{uv}(s) \neq 0$, and consequently, $\nu_{uv}(s) \neq 0$, holds. Along the same lines, one can also show that indeed, $q_{uv}(s) > 0$, and consequently, $\nu_{uv}(s) > 0$, holds (otherwise, we would again obtain a contradiction to condition (24)).

The other cases are analogous adaptations of this discussion and the findings above.

□

### E.5 Proof of Lemma C.8

**Lemma** (Lemma C.8)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. If there exists a self-loop time $\delta t_{uv} \in \mathbb{R}_{\neq 0}$ w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$, the following properties hold:*

1. *There exists no other self-loop time of the same sign (i.e., there exists either exactly one or no positive self-loop time, and there exists exactly one or no negative self-loop time).*
2. *The mean velocity*

$$\overline{\nu}_{uv}(t) := \overline{\nu}_{uv}(t-\delta t_{uv}, t) = \fint_{t-\delta t_{uv}}^{t} \nu_{uv}(s) \; ds = \frac{1}{\delta t_{uv}} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s) \; ds$$

   *during the time interval $[\min\{t-\delta t_{uv}, t\}, \max\{t-\delta t_{uv}, t\}]$ as defined in Theorem C.5 satisfies $\overline{\nu}_{uv}(t) = 0$.*

3. *The flows $q_{uv}(t-\delta t_{uv})$ and $q_{uv}(t)$ linked to the velocity $\nu_{uv}(\cdot) = \frac{q_{vu(\cdot)}}{\alpha_{vu}}$ satisfy*

$$\begin{cases} q_{uv}(t-\delta t_{uv}) \geq 0, \; q_{uv}(t) \leq 0 & \text{if } \delta t_{uv} > 0 \\ q_{uv}(t-\delta t_{uv}) \leq 0, \; q_{uv}(t) \geq 0 & \text{if } \delta t_{uv} < 0 \end{cases}.$$

The proof of Lemma C.8 is very similar to the proof of Lemma C.7 and only requires a few changes. The interested reader can compare both proofs and investigate why the changes are required, and how they induce the different results in Lemma C.8.3 as compared to Lemma C.7.4.

*Proof. 1.:* Let us assume that there exist two different self-loop times $\delta t_{1uv}, \delta t_{2uv} \in \mathbb{R}_{\neq 0}$ with the same sign. If $\delta t_{1uv}, \delta t_{2uv} > 0$ holds, w.l.o.g., let $0 < \delta t_{1uv} < \delta t_{2uv}$ hold. Consequently, for $t' = t - \delta t_{1uv} \in (t-\delta t_{2uv}, t) = (\min\{t-\delta t_{2uv}, t\}, \max\{t-\delta t_{2uv}, t\})$, by (1) by basic results from analysis and (2) condition (23), we obtain

$$\int_{t-\delta t_{2uv}}^{t'} \nu_{uv}(s) \; ds \overset{(1)}{=} \int_{t-\delta t_{2uv}}^{t} \nu_{uv}(s) \; ds - \int_{t-\delta t_{1uv}}^{t} \nu_{uv}(s) \; ds \overset{(2)}{=} 0 - 0 \overset{(1)}{=} 0$$

– a contradiction to condition (20).

If $\delta t_{1uv}, \delta t_{2uv} < 0$ holds, w.l.o.g., let $\delta t_{2uv} < \delta t_{1uv} < 0$ hold. Consequently, for $t' = t - \delta t_{1uv} \in (t, t-\delta t_{2uv}) = (\min\{t-\delta t_{2uv}, t\}, \max\{t-\delta t_{2uv}, t\})$, we can use the same calculation to again obtain a contradiction.

*2.:* By (1) definition of the mean velocity $\overline{\nu}_{uv}(t-\delta t_{uv},t)$ as defined in Theorem C.5 and (2) condition (23), we indeed obtain

$$\overline{\nu}_{uv}(t) := \overline{\nu}_{uv}(t-\delta t_{uv},t) \overset{(1)}{=} \fint_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(1)}{=} \frac{1}{\delta t_{uv}} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(2)}{=} 0.$$

*3.:* If $\delta t_{uv} > 0$ holds, let us assume that $q_{uv}(t-\delta t_{uv}) < 0$, and consequently, $\nu_{uv}(t-\delta t_{uv}) < 0$ holds. Since $\nu_{uv} \in C^0(\mathbb{R},\mathbb{R})$ is continuous, there exists a $t^{'} \in (t-\delta t_{uv},t) = (\min\{t-\delta t_{uv},t\},\max\{t-\delta t_{uv},t\})$ such that $\nu_{uv}(s) < 0$ holds for all $s \in (t-\delta t_{uv},t^{'})$. Consequently, by (1) basic results from analysis, we obtain

$$\int_{t-\delta t_{uv}}^{t^{'}} \underbrace{\nu_{uv}(s)}_{<0}\ ds \overset{(1)}{<} 0$$

– a contradiction to condition (24).

Similarly, if $\delta t_{uv} > 0$ holds, let us assume that $q_{uv}(t) > 0$, and consequently, $\nu_{uv}(t) > 0$ holds. Since $\nu_{uv} \in C^0(\mathbb{R},\mathbb{R})$ is continuous, there exists a $t^{'} \in (t-\delta t_{uv},t) = (\min\{t-\delta t_{uv},t\},\max\{t-\delta t_{uv},t\})$ such that $\nu_{uv}(s) > 0$ holds for all $s \in (t^{'},t)$. Consequently, by (1) basic results from analysis and (2) condition (23), we obtain

$$\int_{t-\delta t_{uv}}^{t^{'}} \nu_{uv}(s)\ ds \overset{(1)}{=} \underbrace{\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds}_{\overset{(2)}{=}0} \underbrace{- \underbrace{\int_{t^{'}}^{t} \underbrace{\nu_{uv}(s)}_{>0}\ ds}_{\overset{(1)}{>}0}}_{<0} < 0$$

– again a contradiction to condition (24).

In contrast, if $\delta t_{uv} < 0$ holds, let us assume that $q_{uv}(t-\delta t_{uv}) > 0$, and consequently, $\nu_{uv}(t-\delta t_{uv}) > 0$ holds. Since $\nu_{uv} \in C^0(\mathbb{R},\mathbb{R})$ is continuous, there exists a $t^{'} \in (t,t-\delta t_{uv}) = (\min\{t-\delta t_{uv},t\},\max\{t-\delta t_{uv},t\})$ such that $\nu_{uv}(s) > 0$ holds for all $s \in (t^{'},t-\delta t_{uv})$. Consequently, by (1) basic results from analysis, we obtain

$$\int_{t-\delta t_{uv}}^{t^{'}} \nu_{uv}(s)\ ds \overset{(1)}{=} - \underbrace{\int_{t^{'}}^{t-\delta t_{uv}} \underbrace{\nu_{uv}(s)}_{>0}\ ds}_{\overset{(1)}{>}0} < 0$$

– again a contradiction to condition (24).

Similarly, if $\delta t_{uv} < 0$ holds, let us assume that $q_{uv}(t) < 0$, and consequently, $\nu_{uv}(t) < 0$ holds. Since $\nu_{uv} \in C^0(\mathbb{R},\mathbb{R})$ is continuous, there exists a $t^{'} \in (t,t-\delta t_{uv}) = (\min\{t-\delta t_{uv},t\},\max\{t-\delta t_{uv},t\})$ such that $\nu_{uv}(s) < 0$ holds for all $s \in (t,t^{'})$. Consequently, by (1) basic results from analysis and (2) condition (23), we obtain

$$\begin{aligned}\int_{t-\delta t_{uv}}^{t^{'}} \nu_{uv}(s)\ ds &\overset{(1)}{=} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds - \int_{t^{'}}^{t} \nu_{uv}(s)\ ds \\ &\overset{(1)}{=} \underbrace{\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds}_{\overset{(2)}{=}0} + \underbrace{\underbrace{\int_{t}^{t^{'}} \underbrace{\nu_{uv}(s)}_{<0}\ ds}_{\overset{(1)}{<}0}}_{<0} < 0\end{aligned}$$

– again a contradiction to condition (24).

*Remark* E.2. Note that if one of the flow rates discussed above are zero, that they can only be zero in this point of time, but not on a whole sub-interval of the time interval $[\min\{t-\delta t_{uv}, t\}, \max\{t-\delta t_{uv}, t\}]$.

More precisely, if $\delta t_{uv} > 0$ holds, let us assume that there exists an $\epsilon > 0$ such that for all $s \in [t-\delta t_{uv}, t-\delta t_{uv}+\epsilon)$, $q_{uv}(s) = 0$, and consequently, $\nu_{uv}(s) = 0$, holds. Thus, for $t^{'} = \min\{t-\delta t_{uv}+\epsilon, t\} \in (t-\delta t_{uv}, t) = (\min\{t-\delta t_{uv}, t\}, \max\{t-\delta t_{uv}, t\})$, by (1) basic results from analysis, we obtain

$$\int_{t-\delta t_{uv}}^{t^{'}} \underbrace{\nu_{uv}(s)}_{=0} \; ds \overset{(1)}{=} 0$$

– a contradiction to condition (24). In combination with the findings above, therefore, $q_{uv}(t-\delta t_{uv}) \geq 0$ needs to hold, and if the (unlikely) case $q_{uv}(t-\delta t_{uv}) = 0$ indeed applies, we observe that for all $\epsilon > 0$ there exists an $s \in [t-\delta t_{uv}, t-\delta t_{uv}+\epsilon)$ such that $q_{uv}(s) \neq 0$, and consequently, $\nu_{uv}(s) \neq 0$, holds. Along the same lines, one can also show that indeed, $q_{uv}(s) > 0$, and consequently, $\nu_{uv}(s) > 0$, holds (otherwise, we would again obtain a contradiction to condition (24)).

The other cases are analogous adaptations of this discussion and the findings above.

□

### E.6 Proof of Lemma C.9

**Lemma** (Lemma C.9)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. $\delta t_{uv} \in \mathbb{R}$ is an inverse pass-through time w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$ iff $\delta t_{vu} := \delta t_{uv} \in \mathbb{R}$ is a pass-through time w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$*

*Proof.* "⇒" If $\delta t_{uv} \in \mathbb{R}$ is an inverse pass-through time w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$, by (1) the fact that the geometric pipe features are symmetric in the sense that $l_{vu} = l_{uv}$ and $\alpha_{vu} = \alpha_{uv}$ holds for all $v \in V$ and $u \in \mathcal{N}(v)$, (2) definition of the flow velocity $\nu_{vu}(\cdot) = \frac{q_{vu}(\cdot)}{\alpha_{vu}}$, (3) the conservation of flows (cf. Eq. (3)) and (4) condition (21), we obtain

$$\int_{t-\delta t_{uv}}^{t} \nu_{vu}(s) \; ds \overset{(1,2,3)}{=} -\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s) \; ds \overset{(4)}{=} -(-l_{uv}) \overset{(1)}{=} l_{vu},$$

that is, condition (19) is satisfied for $\delta t_{vu} := \delta t_{uv}$ and w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$.

Similarly, by additionally using (5) condition (22), we obtain

$$\int_{t-\delta t_{uv}}^{t^{'}} \nu_{vu}(s) \; ds \overset{(1,2,3)}{=} -\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s) \; ds \overset{(5)}{\in} (0, l_{uv}) \overset{(1)}{=} (0, l_{vu}),$$

that is, condition (20) is satisfied for $\delta t_{vu} := \delta t_{uv}$ and w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$. Therefore, $\delta t_{vu} := \delta t_{uv} \in \mathbb{R}$ is a pass-through time w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$.

"⇐" If $\delta t_{vu} := \delta t_{uv} \in \mathbb{R}$ is a pass-through time w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$, by (1) the fact that the geometric pipe features are symmetric in the sense that $l_{vu} = l_{uv}$ and $\alpha_{vu} = \alpha_{uv}$ holds for all $v \in V$ and $u \in \mathcal{N}(v)$, (2) definition of the flow velocity $\nu_{uv}(\cdot) = \frac{q_{uv}(\cdot)}{\alpha_{uv}}$, (3) the conservation of flows (cf. Eq. (3)) and (4) condition (19), we obtain

$$\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s) \; ds \overset{(1,2,3)}{=} -\int_{t-\delta t_{uv}}^{t} \nu_{vu}(s) \; ds \overset{(4)}{=} -(l_{vu}) \overset{(1)}{=} -l_{uv},$$

that is, condition (21) is satisfied for $\delta t_{uv} := \delta t_{vu}$ and w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$.

Similarly, by additionally using (5) condition (20), we obtain

$$\int_{t-\delta t_{uv}}^{t^{'}} \nu_{uv}(s) \; ds \overset{(1,2,3)}{=} -\int_{t-\delta t_{uv}}^{t} \nu_{vu}(s) \; ds \overset{(5)}{\in} (-l_{vu}, 0) \overset{(1)}{=} (-l_{uv}, 0),$$

that is, condition (22) is satisfied for $\delta t_{uv} := \delta t_{vu}$ and w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$. Therefore, $\delta t_{uv} := \delta t_{vu} \in \mathbb{R}$ is an inverse pass-through time w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$. □

### E.7 Proof of Lemma C.10

**Lemma** (Lemma C.10)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. $\delta t_{uv} \in \mathbb{R}$ is an inverse self-loop time w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$ iff $\delta t_{vu} := \delta t_{uv} \in \mathbb{R}$ is a self-loop time w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$*

The proof of Lemma C.10 is very similar to the proof of Lemma C.9 and only requires a few changes. The interested reader can compare both proofs and investigate why the changes are required, and how they induce the different results in Lemma C.10 as compared to Lemma C.10.

*Proof.* "⇒" If $\delta t_{uv} \in \mathbb{R}$ is an inverse self-loop time w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$, by (1) the fact that the geometric pipe features are symmetric in the sense that $l_{vu} = l_{uv}$ and $\alpha_{vu} = \alpha_{uv}$ holds for all $v \in V$ and $u \in \mathcal{N}(v)$, (2) definition of the flow velocity $\nu_{vu}(\cdot) = \frac{q_{vu(\cdot)}}{\alpha_{vu}}$, (3) the conservation of flows (cf. Eq. (3)) and (4) condition (25), we obtain

$$\int_{t-\delta t_{uv}}^{t} \nu_{vu}(s)\ ds \overset{(1,2,3)}{=} -\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(4)}{=} -0 = 0,$$

that is, condition (23) is satisfied for $\delta t_{vu} := \delta t_{uv}$ and w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$.

Similarly, by additionally using (5) condition (26), we obtain

$$\int_{t-\delta t_{uv}}^{t'} \nu_{vu}(s)\ ds \overset{(1,2,3)}{=} -\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(5)}{\in} (0, l_{uv}) \overset{(1)}{=} (0, l_{vu}),$$

that is, condition (24) is satisfied for $\delta t_{vu} := \delta t_{uv}$ and w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$. Therefore, $\delta t_{vu} := \delta t_{uv} \in \mathbb{R}$ is a self-loop time w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$.

"⇐" If $\delta t_{vu} := \delta t_{uv} \in \mathbb{R}$ is a self-loop time w.r.t. $e_{vu} \in E$ and $t \in \mathbb{R}$, by (1) the fact that the geometric pipe features are symmetric in the sense that $l_{vu} = l_{uv}$ and $\alpha_{vu} = \alpha_{uv}$ holds for all $v \in V$ and $u \in \mathcal{N}(v)$, (2) definition of the flow velocity $\nu_{uv}(\cdot) = \frac{q_{uv(\cdot)}}{\alpha_{uv}}$, (3) the conservation of flows (cf. Eq. (3)) and (4) condition (23), we obtain

$$\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(1,2,3)}{=} -\int_{t-\delta t_{uv}}^{t} \nu_{vu}(s)\ ds \overset{(4)}{=} -(0) = 0,$$

that is, condition (25) is satisfied for $\delta t_{uv} := \delta t_{vu}$ and w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$.

Similarly, by additionally using (5) condition (24), we obtain

$$\int_{t-\delta t_{uv}}^{t'} \nu_{uv}(s)\ ds \overset{(1,2,3)}{=} -\int_{t-\delta t_{uv}}^{t} \nu_{vu}(s)\ ds \overset{(5)}{\in} (-l_{vu}, 0) \overset{(1)}{=} (-l_{uv}, 0),$$

that is, condition (26) is satisfied for $\delta t_{uv} := \delta t_{vu}$ and w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$. Therefore, $\delta t_{uv} := \delta t_{vu} \in \mathbb{R}$ is an inverse self-loop time w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$. □

### E.8 Proof of Theorem C.11

**Theorem** (Theorem C.11)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. If the function $c_{uv}$ obeys Equation (2) and there exists a pass-through time $\delta t_{uv} \in \mathbb{R}_{\neq 0}$ w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$, we obtain*

$$c_{uv}(t, l_{uv}) = c_{uv}(t - \delta t_{uv}, 0) = c_{uv}\left(t - \frac{l_{uv}}{\overline{\nu}_{uv}(t)}, 0\right).$$

*Even more, the concentrations $c_{uv}(\cdot, l_{uv})$ and $c_{vu}(\cdot, l_{vu})$ at the end $z = l_{uv} = l_{vu}$ of the edges $e_{uv} \in E$ and $e_{vu} \in E$ are connected to the concentrations $c_u$ and $c_v$ of the nodes of that edge, respectively, by*

$$\begin{cases} c_{uv}(t, l_{uv}) = c_u(t - \delta t_{uv}) & \text{if } \delta t_{uv} > 0 \\ c_{vu}(t - \delta t_{uv}, l_{vu}) = c_v(t) & \text{if } \delta t_{uv} < 0 \end{cases}.$$

*Proof.* The main idea of this proof is to use the continuity of the function $c_{uv} \in C^0(\mathbb{R} \times [0, l_{uv}], \mathbb{R}_{\geq 0})$ to transfer the results from the advective transport along edges, i.e., Theorem C.5, to the ends of the edge $e_{uv} \in E$, i.e., to the functions $c_u, c_v \in C^1(\mathbb{R}, \mathbb{R}_{\geq 0})$.

The crucial part is that we would like to apply Theorem C.5 for $t_0 = t - \delta t_{uv}$, $z_0 = 0$ and $t \in \mathbb{R}$ as given. However, the first problem is that $z_0 = 0 \notin \Omega_{uv} = (0, l_{uv})$ does not satisfy the condition of Theorem C.5. Moreover, since for this choice,

$$z_0 + \int_{t_0}^{t_0} \nu_{uv}(s)\ ds = z_0 + 0 = 0 \quad \text{and} \quad z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds = z_0 + l_{uv} = l_{uv}$$

hold, $t \notin T_{t_0,z_0}$ does also not satisfy the condition of Theorem C.5. Intuitively, we first have to transfer the information from the boundary of the edge $e_{uv}$ to the inside of the edge using the continuity of the function $c_{uv} \in C^1(\mathbb{R} \times [0, l_{uv}], \mathbb{R}_{\geq 0})$. Afterwards, we can use Theorem C.5 for slightly modified $z_0$ and $t$.

To do so, we define the two functions

$$\begin{aligned} z :&(\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \longrightarrow \mathbb{R},\ \epsilon \longmapsto z(\epsilon) := \int_{t-\delta t_{uv}}^{t-\epsilon} \nu_{uv}(s)\ ds \text{ and} \\ \tilde{z} :&(\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \longrightarrow \mathbb{R},\ \epsilon \longmapsto \tilde{z}(e) := \frac{l_{uv} - z(\epsilon)}{2}. \end{aligned}$$

We first have to prove some technical details regarding these functions (*Step 1*) in order to be able to apply Theorem C.5 as illustrated above (*Step 2*). Afterwards, we can use the results from Lemma C.7 about the sign of the flow at time $t - \delta t_{uv}$ in dependence of the sign of the pass-through time $\delta t_{uv}$ in combination with the local inflow boundary conditions defined through Equation (5) to be able to transfer the results from the advective transport along edges to the end of the pipes (*Step 3*).

*Step 1: The functions $z$ and $\tilde{z}$ satisfy the following properties:*

1. $z$, $\tilde{z}$ and $z + \tilde{z}$ are continuous.
2. $\text{im}(z) \subset (0, l_{uv})$, $\text{im}(\tilde{z}) \subset (0, \frac{1}{2} l_{uv}) \subset (0, l_{uv})$ and $\text{im}(z + \tilde{z}) \subset (\frac{1}{2}, l_{uv}) \subset (0, l_{uv})$.
3. $\lim_{\epsilon \to 0} z(\epsilon) = l_{uv}$, $\lim_{\epsilon \to 0} \tilde{z}(\epsilon) = 0$ and $\lim_{\epsilon \to 0} z(\epsilon) + \tilde{z}(\epsilon) = l_{uv}$.

Note that the limit is to be understood for $\epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})$ in the domain of $z$ and $\tilde{z}$, i.e., if $\delta t_{uv} > 0$, $(\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) = (0, \delta t_{uv})$ holds, and the limit $\epsilon \to 0$ is to be understood as the limit $\epsilon \searrow 0, \epsilon < \delta t_{uv}$. In contrast, if $\delta t_{uv} < 0$, $(\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) = (\delta t_{uv}, 0)$ holds, and the limit $\epsilon \to 0$ is to be understood as the limit $\epsilon \nearrow 0, \epsilon > \delta t_{uv}$.

*1.:* By the fundamental theorem of calculus, the function $t \mapsto \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds$ is differentiable and therefore, continuous. Therefore, as a composition of continuous functions, the functions $z$, $\tilde{z}$ and $z + \tilde{z}$ are continuous.

*2.:* By (1) basic transformations, (2) definition of $z$, $\tilde{z}$ and $z + \tilde{z}$, respectively, and (3) condition (20), we obtain

$$\text{im}(z) \overset{(1)}{=} \{z(\epsilon) \mid \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})\}$$

$$\overset{(2)}{=} \left\{ \int_{t-\delta t_{uv}}^{t-\epsilon} \nu_{uv}(s)\ ds \ \middle|\ \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \right\}$$
$$\overset{(1)}{=} \left\{ \int_{t-\delta t_{uv}}^{t-\epsilon} \nu_{uv}(s)\ ds \ \middle|\ t-\epsilon \in (\min\{t, t-\delta t_{uv}\}, \max\{t, t-\delta t_{uv}\}) \right\}$$
$$\overset{(3)}{\subset} (0, l_{uv}).$$

By additionally using that therefore, (4) $z(\epsilon) \in (0, l_{uv})$ holds for all $\epsilon \in (\min\{0, \delta t_{uv}\},\ \max\{0, \delta t_{uv}\})$, we obtain

$$\begin{aligned}
\operatorname{im}(\tilde{z}) &\overset{(1)}{=} \{\tilde{z}(\epsilon) \mid \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \\
&\overset{(2)}{=} \left\{ \frac{l_{uv} - z(\epsilon)}{2} \ \middle|\ \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \right\} \\
&\overset{(4)}{\subset} (0, \tfrac{1}{2} l_{uv}) \quad \text{and} \\
\operatorname{im}(z + \tilde{z}) &\overset{(1)}{=} \{z(\epsilon) + \tilde{z}(\epsilon) \mid \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})\} \\
&\overset{(2)}{=} \left\{ z(\epsilon) + \frac{l_{uv} - z(\epsilon)}{2} \ \middle|\ \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \right\} \\
&\overset{(1)}{=} \left\{ \frac{l_{uv} + z(\epsilon)}{2} \ \middle|\ \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \right\} \\
&\overset{(4)}{\subset} (\tfrac{1}{2} l_{uv}, l_{uv}).
\end{aligned}$$

*3.:* By (1) continuity of $z$, $\tilde{z}$ and $z + \tilde{z}$, respectively, (2) definition of $z$, $\tilde{z}$ and $z + \tilde{z}$, respectively and (3) condition (19), we obtain

$$\begin{aligned}
\lim_{\epsilon \to 0} z(\epsilon) &\overset{(1)}{=} z(0) \overset{(2)}{=} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(3)}{=} l_{uv}, \\
\lim_{\epsilon \to 0} \tilde{z}(\epsilon) &\overset{(1)}{=} \tilde{z}(0) \overset{(2)}{=} \frac{l_{uv} - z(0)}{2} \overset{(2,3)}{=} 0, \\
\lim_{\epsilon \to 0} z(\epsilon) + \tilde{z}(\epsilon) &\overset{(1)}{=} z(0) + \tilde{z}(0) \overset{(2,3)}{=} l_{uv}.
\end{aligned}$$

*Step 2: $c_{uv}(t, l_{uv}) = c_{uv}(t - \delta t_{uv}, 0) = c_{uv}(t - \frac{l_{uv}}{\overline{\nu}_{uv}(t)}, 0)$ holds.*

For each $\epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})$, we can now choose $t_0 = t - \delta t_{uv} \in \mathbb{R}$ and – as shown in *Step 1* – $z_0 = \tilde{z}(\epsilon) \in (0, l_{uv})$. Consequently, for each $\epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})$, by (1) the choices of $t_0 = t - \delta t_{uv} \in \mathbb{R}$ and $z_0 = \tilde{z}(\epsilon) \in (0, l_{uv})$, (2) the definition of $z$ and $\tilde{z}$ from *Step 1* and (3) the results from *Step 1*, we observe that

$$z_0 + \int_{t_0}^{t-\epsilon} \nu_{uv}(s)\ ds \overset{(1)}{=} \tilde{z}(\epsilon) + \int_{t-\delta t_{uv}}^{t-\epsilon} \nu_{uv}(s)\ ds \overset{(2)}{=} \tilde{z}(\epsilon) + z(\epsilon) \overset{(3)}{\in} (0, l_{uv}),$$

and therefore, $t - \epsilon \in T_{t_0, z_0} = T_{t-\delta t_{uv}, \tilde{z}(\epsilon)}$ holds.[18]

Therefore, by (1) the results from *Step 1*, (2) the continuity of the function $c_{uv} \in C^1(\mathbb{R} \times [0, l_{uv}], \mathbb{R}_{\geq 0})$, (3) definition of $z$ and (4) Theorem C.5 applied to $t_0 = t - \delta t_{uv}$, $z_0 = \tilde{z}(\epsilon) \in (0, l_{uv})$ and $t - \epsilon \in T_{t_0, z_0} = T_{t-\delta t_{uv}, \tilde{z}(\epsilon)}$, we obtain

$$c_{uv}(t, l_{uv}) \overset{(1)}{=} c_{uv}\left( \lim_{\epsilon \to 0} t - \epsilon, \lim_{\epsilon \to 0} z(\epsilon) + \tilde{z}(\epsilon) \right)$$

[18]Note that all the technical work of constructing $z$ and $\tilde{z}$ in *Step 1* is because we would actually like to choose $z_0 = 0$ and $t \in \mathbb{R}$ as given, however, this $z_0$ and this $t$ do not satisfy the requirements of Theorem C.5.

$$
\begin{aligned}
&\overset{(2)}{=} \lim_{\epsilon\to 0} c_{uv}\left(t-\epsilon, z(\epsilon)+\tilde{z}(\epsilon)\right)\\
&\overset{(3)}{=} \lim_{\epsilon\to 0} c_{uv}\Bigg(t-\epsilon\ ,\ \underbrace{\underbrace{\tilde{z}(\epsilon)}_{\overset{(1)}{\in}(0,l_{uv})} + \underbrace{\int_{t-\delta t_{uv}}^{t-\epsilon} \nu_{uv}(s)\ ds}_{=z(\epsilon)\overset{(1)}{\in}(0,l_{uv})}}_{=z(\epsilon)+\tilde{z}(\epsilon)\overset{(1)}{\in}(0,l_{uv})}\Bigg)\\
&\overset{(4)}{=} \lim_{\epsilon\to 0} c_{uv}\left(t-\delta t_{uv}, \tilde{z}(\epsilon)\right)\\
&\overset{(2)}{=} c_{uv}\left(t-\delta t_{uv}, \lim_{\epsilon\to 0}\tilde{z}(\epsilon)\right)\\
&\overset{(1)}{=} c_{uv}(t-\delta t_{uv}, 0).
\end{aligned}
$$

Finally, as seen in Lemma C.7.3, $\delta t_{uv} = \frac{l_{uv}}{\overline{\nu}_{uv}(t)}$ holds, and we can conclude

$$
c_{uv}(t-\delta t_{uv}, 0) = c_{uv}\left(t - \frac{l_{uv}}{\overline{\nu}_{uv}(t)}\right).
$$

*Step 3: If $\delta t_{uv} > 0$ holds, $c_{uv}(t, l_{uv}) = c_u(t-\delta t_{uv})$ holds. If $\delta t_{uv} < 0$ holds, $c_{vu}(t-\delta t_{uv}, l_{vu}) = c_v(t)$ holds.*

If $\delta t_{uv} > 0$ holds, by (1) *Step 2* and (2) Equation (5) together with the fact that by Lemma C.7.4, $q_{uv}(t-\delta t_{uv}) \geq 0$ holds, we obtain

$$
c_{uv}(t, l_{uv}) \overset{(1)}{=} c_{uv}(t-\delta t_{uv}, 0) \overset{(2)}{=} c_u(t-\delta t_{uv}).
$$

If instead, $\delta t_{uv} < 0$ holds, by (1) Equation (18) for $z = l_{vu}$, (2) *Step 2* and (3) Equation (5) together with the fact that by the conservation of flows (Eq. (3)) and Lemma C.7.4, $q_{vu}(t) = -q_{uv}(t) \geq 0$ holds, we obtain

$$
c_{vu}(t-\delta t_{uv}, l_{vu}) \overset{(1)}{=} c_{uv}(t-\delta t_{uv}, 0) \overset{(2)}{=} c_{uv}(t, l_{uv}) \overset{(1)}{=} c_{vu}(t, 0) \overset{(3)}{=} c_v(t).
$$

□

### E.9 Proof of Theorem C.12

**Theorem** (Theorem C.12)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. If the function $c_{uv}$ obeys Equation* (2) *and there exists a self-loop time $\delta t_{uv} \in \mathbb{R}_{\neq 0}$ w.r.t. $e_{uv} \in E$ and $t \in \mathbb{R}$, we obtain*

$$
c_{uv}(t, 0) = c_{uv}(t-\delta t_{uv}, 0).
$$

*Even more, the concentration $c_{vu}(\cdot, l_{vu})$ at the end $z = l_{vu} = l_{uv}$ of the edge $e_{vu} \in E$ is connected to the concentration $c_u$ of the node of that edge by*

$$
\begin{cases} c_{vu}(t, l_{vu}) = c_u(t-\delta t_{uv}) & \text{if } \delta t_{uv} > 0 \\ c_{vu}(t-\delta t_{uv}, l_{vu}) = c_u(t) & \text{if } \delta t_{uv} < 0 \end{cases}.
$$

The proof of Theorem C.12 is very similar to the proof of Theorem C.11 and only requires a few changes. The interested reader can compare both proofs and investigate why the changes are required, and how they induce the different results in Theorem C.12 as compared to Theorem C.11.

*Proof.* The main idea of this proof is to use the continuity of the function $c_{uv} \in C^1(\mathbb{R} \times [0, l_{uv}], \mathbb{R}_{\geq 0})$ to transfer the results from the advective transport along edges, i.e., Theorem C.5, to the end of the edge $e_{uv} \in E$, i.e., to the function $c_u \in C^1(\mathbb{R}, \mathbb{R}_{\geq 0})$.

The crucial part is that we would like to apply Theorem C.5 for $t_0 = t - \delta t_{uv}$, $z_0 = 0$ and $t \in \mathbb{R}$ as given. However, the first problem is that $z_0 = 0 \notin \Omega_{uv} = (0, l_{uv})$ does not satisfy the condition of Theorem C.5. Moreover, since for this choice,

$$z_0 + \int_{t_0}^{t_0} \nu_{uv}(s)\ ds = z_0 + 0 = 0 \quad \text{and} \quad z_0 + \int_{t_0}^{t} \nu_{uv}(s)\ ds = z_0 + 0 = 0$$

hold, $t \notin T_{t_0,z_0}$ does also not satisfy the condition of Theorem C.5. Intuitively, we first have to transfer the information from the boundary of the edge $e_{uv}$ to the inside of the edge using the continuity of the function $c_{uv} \in C^1(\mathbb{R} \times [0, l_{uv}], \mathbb{R}_{\geq 0})$. Afterwards, we can use Theorem C.5 for slightly modified $z_0$ and $t$.

To do so, we define the two functions

$$\begin{aligned} z :&(\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \longrightarrow \mathbb{R},\ \epsilon \longmapsto z(\epsilon) := \int_{t-\delta t_{uv}}^{t-\epsilon} \nu_{uv}(s)\ ds \text{ and} \\ \tilde{z} :&(\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \longrightarrow \mathbb{R},\ \epsilon \longmapsto \tilde{z}(e) := \frac{z(\epsilon)}{2}. \end{aligned}$$

We first have to prove some technical details regarding these functions (*Step 1*) in order to be able to apply Theorem C.5 as illustrated above (*Step 2*). Afterwards, we can use the results from Lemma C.8 about the sign of the flow at time $t - \delta t_{uv}$ in dependence of the sign of the self-loop time $\delta t_{uv}$ in combination with the local inflow boundary conditions defined through Equation (5) to be able to transfer the results from the advective transport along edges to the end of the pipe (*Step 3*).

*Step 1: The functions $z$ and $\tilde{z}$ satisfy the following properties:*

1. $z$, $\tilde{z}$ and $z + \tilde{z}$ are continuous.
2. $\text{im}(z) \subset (0, l_{uv})$, $\text{im}(\tilde{z}) \subset (0, \frac{1}{2} l_{uv}) \subset (0, l_{uv})$ and $\text{im}(z + \tilde{z}) \subset (0, \frac{3}{2} l_{uv}) \subset (0, l_{uv})$.
3. $\lim_{\epsilon \to 0} z(\epsilon) = 0$, $\lim_{\epsilon \to 0} \tilde{z}(\epsilon) = 0$ and $\lim_{\epsilon \to 0} z(\epsilon) + \tilde{z}(\epsilon) = 0$.

Note that the limit is to be understood for $\epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})$ in the domain of $z$ and $\tilde{z}$, i.e., if $\delta t_{uv} > 0$, $(\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) = (0, \delta t_{uv})$ holds, and the limit $\epsilon \to 0$ is to be understood as the limit $\epsilon \searrow 0, \epsilon < \delta t_{uv}$. In contrast, if $\delta t_{uv} < 0$, $(\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) = (\delta t_{uv}, 0)$ holds, and the limit $\epsilon \to 0$ is to be understood as the limit $\epsilon \nearrow 0, \epsilon > \delta t_{uv}$.

*1.:* By the fundamental theorem of calculus, the function $t \mapsto \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds$ differentiable and therefore, continuous. Therefore, as compositions of continuous functions, the functions $z$, $\tilde{z}$ and $z + \tilde{z}$ are continuous.

*2.:* By (1) basic transformations, (2) definition of $z$, $\tilde{z}$ and $z + \tilde{z}$, respectively, and (3) condition (24), we obtain

$$\begin{aligned} \text{im}(z) &\overset{(1)}{=} \{z(\epsilon) \mid \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})\} \\ &\overset{(2)}{=} \left\{ \int_{t-\delta t_{uv}}^{t-\epsilon} \nu_{uv}(s)\ ds \;\middle|\; \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \right\} \\ &\overset{(1)}{=} \left\{ \int_{t-\delta t_{uv}}^{t-\epsilon} \nu_{uv}(s)\ ds \;\middle|\; t - \epsilon \in (\min\{t, t - \delta t_{uv}\}, \max\{t, t - \delta t_{uv}\}) \right\} \\ &\overset{(3)}{\subset} (0, l_{uv}). \end{aligned}$$

By additionally using that therefore, (4) $z(\epsilon) \in (0, l_{uv})$ holds for all $\epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})$, we obtain

$$\text{im}(\tilde{z}) \overset{(1)}{=} \{\tilde{z}(\epsilon) \mid \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})$$

$$
\begin{aligned}
&\overset{(2)}{=} \left\{ \frac{z(\epsilon)}{2} \;\middle|\; \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \right\} \\
&\overset{(4)}{\subset} (0, \tfrac{1}{2} l_{uv}) \quad \text{and} \\
\operatorname{im}(z + \tilde{z}) &\overset{(1)}{=} \{z(\epsilon) + \tilde{z}(\epsilon) \mid \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})\} \\
&\overset{(2)}{=} \left\{ z(\epsilon) + \frac{z(\epsilon)}{2} \;\middle|\; \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \right\} \\
&\overset{(1)}{=} \left\{ \frac{3}{2} z(\epsilon) \;\middle|\; \epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\}) \right\} \\
&\overset{(4)}{\subset} (0, \tfrac{3}{2} l_{uv}).
\end{aligned}
$$

*3.:* By (1) continuity of $z$, $\tilde{z}$ and $z + \tilde{z}$, respectively, (2) definition of $z$, $\tilde{z}$ and $z + \tilde{z}$, respectively and (3) condition (23), we obtain

$$
\begin{aligned}
\lim_{\epsilon \to 0} z(\epsilon) &\overset{(1)}{=} z(0) \overset{(2)}{=} \int_{t - \delta t_{uv}}^{t} \nu_{uv}(s)\, ds \overset{(3)}{=} 0, \\
\lim_{\epsilon \to 0} \tilde{z}(\epsilon) &\overset{(1)}{=} \tilde{z}(0) \overset{(2)}{=} \frac{z(0)}{2} \overset{(2,3)}{=} 0, \\
\lim_{\epsilon \to 0} z(\epsilon) + \tilde{z}(\epsilon) &\overset{(1)}{=} z(0) + \tilde{z}(0) \overset{(2,3)}{=} 0.
\end{aligned}
$$

*Step 2: $c_{uv}(t, 0) = c_{uv}(t - \delta t_{uv}, 0)$ holds.*

For each $\epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})$, we can now choose $t_0 = t - \delta t_{uv} \in \mathbb{R}$ and – as shown in *Step 1* – $z_0 = \tilde{z}(\epsilon) \in (0, l_{uv})$. Consequently, for each $\epsilon \in (\min\{0, \delta t_{uv}\}, \max\{0, \delta t_{uv}\})$, by (1) the choices of $t_0 = t - \delta t_{uv} \in \mathbb{R}$ and $z_0 = \tilde{z}(\epsilon) \in (0, l_{uv})$, (2) the definition of $z$ and $\tilde{z}$ from *Step 1* and (3) the results from *Step 1*, we observe that

$$
z_0 + \int_{t_0}^{t-\epsilon} \nu_{uv}(s)\, ds \overset{(1)}{=} \tilde{z}(\epsilon) + \int_{t - \delta t_{uv}}^{t-\epsilon} \nu_{uv}(s)\, ds \overset{(2)}{=} \tilde{z}(\epsilon) + z(\epsilon) \overset{(3)}{\in} (0, l_{uv}),
$$

and therefore, $t - \epsilon \in T_{t_0, z_0} = T_{t - \delta t_{uv}, \tilde{z}(\epsilon)}$ holds.[19]

Therefore, by (1) the results from *Step 1*, (2) the continuity of the function $c_{uv} \in C^1(\mathbb{R} \times [0, l_{uv}], \mathbb{R}_{\geq 0})$, (3) definition of $z$ and (4) Theorem C.5 applied to $t_0 = t - \delta t_{uv}$, $z_0 = \tilde{z}(\epsilon) \in (0, l_{uv})$ and $t - \epsilon \in T_{t_0, z_0} = T_{t - \delta t_{uv}, \tilde{z}(\epsilon)}$, we obtain

$$
\begin{aligned}
c_{uv}(t, 0) &\overset{(1)}{=} c_{uv}\left( \lim_{\epsilon \to 0} t - \epsilon, \lim_{\epsilon \to 0} z(\epsilon) + \tilde{z}(\epsilon) \right) \\
&\overset{(2)}{=} \lim_{\epsilon \to 0} c_{uv}\left( t - \epsilon, z(\epsilon) + \tilde{z}(\epsilon) \right) \\
&\overset{(3)}{=} \lim_{\epsilon \to 0} c_{uv}\Bigg( t - \epsilon \;, \underbrace{\underbrace{\tilde{z}(\epsilon)}_{\overset{(1)}{\in} (0, l_{uv})} + \underbrace{\int_{t - \delta t_{uv}}^{t-\epsilon} \nu_{uv}(s)\, ds}_{= z(\epsilon) \overset{(1)}{\in} (0, l_{uv})}}_{= z(\epsilon) + \tilde{z}(\epsilon) \overset{(1)}{\in} (0, l_{uv})} \Bigg) \\
&\overset{(4)}{=} \lim_{\epsilon \to 0} c_{uv}\left( t - \delta t_{uv}, \tilde{z}(\epsilon) \right) \\
&\overset{(2)}{=} c_{uv}\left( t - \delta t_{uv}, \lim_{\epsilon \to 0} \tilde{z}(\epsilon) \right)
\end{aligned}
$$

[19] Note that all the technical work of constructing $z$ and $\tilde{z}$ in *Step 1* is because we would actually like to choose $z_0 = 0$ and $t \in \mathbb{R}$ as given; however, this $z_0$ and this $t$ do not satisfy the requirements of Theorem C.5.

$$\overset{(1)}{=} c_{uv}(t - \delta t_{uv}, 0).$$

*Step 3: If $\delta t_{uv} > 0$ holds, $c_{vu}(t, l_{vu}) = c_u(t - \delta t_{uv})$ holds. If $\delta t_{uv} < 0$ holds, $c_{vu}(t - \delta t_{uv}, l_{vu}) = c_u(t)$ holds.*

If $\delta t_{uv} > 0$ holds, by (1) Equation (18) for $z = l_{vu}$, (2) *Step 2* and (3) Equation (5) together with the fact that by Lemma C.8.4, $q_{uv}(t - \delta t_{uv}) \geq 0$ holds, we obtain

$$c_{vu}(t, l_{vu}) \overset{(1)}{=} c_{uv}(t, 0) \overset{(2)}{=} c_{uv}(t - \delta t_{uv}, 0) \overset{(3)}{=} c_u(t - \delta t_{uv}).$$

If instead, $\delta t_{uv} < 0$ holds, by (1) Equation (18) for $z = l_{vu}$, (2) *Step 2* and (3) Equation (5) together with the fact that by Lemma C.8.4, $q_{uv}(t) \geq 0$ holds, we obtain

$$c_{vu}(t - \delta t_{uv}, l_{vu}) \overset{(1)}{=} c_{uv}(t - \delta t_{uv}, 0) \overset{(2)}{=} c_{uv}(t, 0) \overset{(3)}{=} c_u(t).$$

□

### E.10 Proof of Theorem C.13

**Theorem** (Theorem C.13)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. If $\nu_{uv}(t) \neq 0$ holds and the set*

$$\begin{aligned} A_{uv}(t) :=& \left\{ \delta t \in (0, \infty) \;\middle|\; \int_{t-\delta t}^{t} \nu_{uv}(s)\, ds \in \{-l_{uv}, 0, l_{uv}\} \right\} \\ =& \left\{ \delta t \in (0, \infty) \;\middle|\; \not{z}(\delta t) \in \{-l_{uv}, 0, l_{uv}\} \right\} \end{aligned}$$

*is not empty, the following properties hold:*

1. *$\delta t_{uv} := \min A_{uv}(t) = \min\{\delta t \in (0, \infty) \mid \not{z}(\delta t) \in \{-l_{uv}, 0, l_{uv}\}\} \in (0, \infty)$ exists.*
2. *$\delta t_{uv}$ is a positive pass-through time, inverse pass-through time, self-loop time or inverse self-loop time. More specifically, $\delta t_{uv}$ is (always w.r.t. $e_{uv}$ and $t$) ...*

   *... an inverse pass-through time* $\iff \not{z}(\delta t_{uv}) = -l_{uv}$,
   *... a self-loop time or an inverse self-loop time* $\iff \not{z}(\delta t_{uv}) = 0$,
   *... a pass-through time* $\iff \not{z}(\delta t_{uv}) = l_{uv}$.

3. *There exists no other positive pass-through time, inverse pass-through time, self-loop time or inverse self-loop time than $\delta t_{uv}$.*

*Proof.* By the fundamental theorem of calculus, the function $t' \mapsto \int_t^{t'} \nu_{uv}(s)\, ds$ is differentiable and therefore, continuous. Therefore, as a composition of continuous functions, the function $\not{z}$ is continuous.

*1:* By assumption, $A_{uv}(t) \neq \emptyset$ holds. Moreover, by definition, the value $0 \in \mathbb{R}$ is a lower bound of $A_{uv}(t) \subset (0, \infty)$. Therefore, it suffices to show that $A_{uv}(t)$ is closed in $\mathbb{R}$; in this case, the minimum of $A_{uv}(t)$ exists.

To do so, let $(\delta t_n)_{n \in \mathbb{N}}$ be a sequence in $A_{uv}(t)$ such that $\delta t_n \to \delta t_0$ holds for $n \to \infty$. We need to show that $\delta t_0 \in A_{uv}(t)$ holds.

*Step 1.1: $\not{z}(\delta t_0) \in \{-l_{uv}, 0, l_{uv}\}$ holds.*

On the one hand, by the continuity of $\not{z}$, we obtain that $\not{z}(\delta t_n) \to \not{z}(\delta t_0)$ holds for $n \to \infty$. One the other hand, since $(\delta t_n)_{n \in \mathbb{N}}$ is a sequence in $A_{uv}(t)$, $(\not{z}(\delta t_n))_{n \in \mathbb{N}}$ is a sequence in $\{-l_{uv}, 0, l_{uv}\}$. Consequently, since $\{-l_{uv}, 0, l_{uv}\}$ is closed in $\mathbb{R}$, the limit $\not{z}(\delta t_0)$ is also an element of $\{-l_{uv}, 0, l_{uv}\}$.

*Step 1.2:* $\delta t_0 > 0$ *holds.*

Since $(\delta t_n)_{n\in\mathbb{N}}$ is a sequence in $A_{uv}(t)$, $\delta t_n > 0$ holds for a $n \in \mathbb{N}$. Consequently, the limit $\delta t_0 = \lim_{n\to\infty} \delta t_n$ satisfies $\delta t_0 \geq 0$, and it suffices to show that $\delta t_0 \neq 0$ holds.

Let us assume that $\delta t_0 = 0$ holds. By the continuity of $\nu_{uv}$ and the assumption that $\nu_{uv}(t) \neq 0$ holds, there exists an $\epsilon > 0$ such that $\nu_{uv}(s) > 0$ or $\nu_{uv}(s) < 0$ holds for all $s \in (t-\epsilon, t+\epsilon)$. W.l.o.g., we can assume that $\epsilon < l_{uv}$ holds (else, we choose a smaller $\epsilon$ that naturally satisfies this condition). Consequently, (1) by basic results from analysis,

$$\text{ƶ}(\delta t) = \int_{t-\delta t}^{t} \underbrace{\nu_{uv}(s)}_{>0 \text{ or } <0} \; ds \overset{(1)}{\neq} 0 \tag{36}$$

holds for all $\delta t \in (0, \epsilon)$. Specifically, since $\delta t_n \to \delta t_0 = 0$ holds for $n \to \infty$, for the $\epsilon > 0$ from above, there exists an $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $|\delta t_n| < \epsilon$ holds; and since $(\delta t_n)_{n\in\mathbb{N}}$ is a sequence in $A_{uv}(t)$, even more, $\delta t_n \in (0, \epsilon)$ holds for all $n \geq n_0$. Consequently, for all $n \geq n_0$, Equation (36) applies to $\delta t_n$; and since again $(\delta t_n)_{n\in\mathbb{N}}$ is a sequence in $A_{uv}(t)$, even more,

$$\text{ƶ}(\delta t_n) \in \{-l_{uv}, l_{uv}\} \tag{37}$$

holds for all $n \geq n_0$.

At the same time, by the continuity of ƶ, we obtain that for the $\epsilon > 0$ from above, there exists an $n_1 \in \mathbb{N}$ such

$$|\text{ƶ}(\delta t_n) - \text{ƶ}(\delta t_0)| < \epsilon \tag{38}$$

holds for all $n \geq n_1$.

Bringing these findings together, by (1) basic transformations, (2) Equation (37) (3) the definition of ƶ (cf. Equation (7)), (4) the assumption that $\delta t_0 = 0$ holds, (5) basic results from analysis, (6) Equation (38) and (7) the choice of $\epsilon$, we obtain

$$l_{uv} \overset{(1)}{=} |\pm l_{uv} - 0| \overset{(2,3,4,5)}{=} |\text{ƶ}(\delta t_n) - \text{ƶ}(\delta t_0)| \overset{(6)}{<} \epsilon \overset{(7)}{<} l_{uv}$$

for all $n \geq \max\{n_0, n_1\}$ – a contradiction.

*2:* For $\delta t_{uv} := \min A_{uv}(t)$, we show an even stronger claim (namely Lemma C.15.1) in several steps.

*Step 2.1:* $\operatorname{im} \text{ƶ}_{|(0,\delta t_{uv})} \subset (-l_{uv}, 0) \sqcup (0, l_{uv})$ *holds.*

Let us assume that there exists a $\delta t \in (0, \delta t_{uv})$ such that $\text{ƶ}(\delta t) \in (-\infty, -l_{uv}] \sqcup \{0\} \sqcup [l_{uv}, \infty)$ holds.
*Case 1:* If $\text{ƶ}(\delta t) \in \{-l_{uv}, 0, l_{uv}\}$ holds, we obtain $\delta t \in A_{uv}(t)$ with $\delta t < \delta t_{uv}$ – a contradiction to the choice of $\delta t_{uv} = \min A_{uv}(t)$ as the minimum of $A_{uv}(t)$.
*Case 2:* If $\text{ƶ}(\delta t) \in (-\infty, -l_{uv})$ holds, by the fact that $\text{ƶ}(0) = 0$ holds and the intermediate value theorem, there exists a $\delta t^{'} \in (0, \delta t)$, such that $\text{ƶ}(\delta t^{'}) = -l_{uv}$ holds. Consequently, we obtain $\delta t^{'} \in A_{uv}(t)$ with $\delta t^{'} < \delta t_{uv}$ – a contradiction to the choice of $\delta t_{uv} = \min A_{uv}(t)$ as the minimum of $A_{uv}(t)$.
*Case 3:* If $\text{ƶ}(\delta t) \in (l_{uv}, \infty)$ holds, by the fact that $\text{ƶ}(0) = 0$ holds and the intermediate value theorem, there exists a $\delta t^{'} \in (0, \delta t)$, such that $\text{ƶ}(\delta t^{'}) = l_{uv}$ holds. Consequently, we obtain $\delta t^{'} \in A_{uv}(t)$ with $\delta t^{'} < \delta t_{uv}$ – a contradiction to the choice of $\delta t_{uv} = \min A_{uv}(t)$ as the minimum of $A_{uv}(t)$.

*Step 2.2: Either* $\operatorname{im} z_{|(0,\delta t_{uv})} \subset (-l_{uv}, 0)$ *or* $\operatorname{im} z_{|(0,\delta t_{uv})} \subset (0, l_{uv})$ *holds.*

By step *Step 2.1*, $\operatorname{im} z_{|(0,\delta t_{uv})} \subset (-l_{uv}, 0) \sqcup (0, l_{uv})$ holds. Let us assume that there exist $\delta t_1, \delta t_2 \in (0, \delta t_{uv})$ such that $\text{ƶ}(\delta t_1) \in (-l_{uv}, 0)$ and $\text{ƶ}(\delta t_2) \in (0, l_{uv})$ holds. Again by the mean value theorem, there exists a

$\delta t \in (\delta t_1, \delta_2)$, such that $\not{z}(\delta t) = 0$ holds. Consequently, we obtain $\delta t \in A_{uv}(t)$ with $\delta t < \delta t_2 < \delta t_{uv}$ – a contradiction to the choice of $\delta t_{uv} = \min A_{uv}(t)$ as the minimum of $A_{uv}(t)$.

*Step 2.3: If additionally, $\not{z}(\delta t_{uv}) = -l_{uv}$ holds, then $\operatorname{im} z_{|(0,\delta t_{uv})} \subset (-l_{uv}, 0)$ holds. If in contrast, additionally, $\not{z}(\delta t_{uv}) = l_{uv}$ holds, then $\operatorname{im} z_{|(0,\delta t_{uv})} \subset (0, l_{uv})$ holds.*

By step *Step 2.2*, either $\operatorname{im} z_{|(0,\delta t_{uv})} \subset (-l_{uv}, 0)$ or $\operatorname{im} z_{|(0,\delta t_{uv})} \subset (0, l_{uv})$ holds. If additionally, $\not{z}(\delta t_{uv}) = -l_{uv}$ holds, let us assume that there exists a $\delta t \in (0, \delta t_{uv})$ such that $\not{z}(\delta t) \in (0, l_{uv})$ holds. Again by the intermediate value theorem, there exists a $\delta t^{'} \in (\delta t, \delta t_{uv})$, such that $\not{z}(\delta t^{'}) = 0$ holds. Consequently, we obtain $\delta t^{'} \in A_{uv}(t)$ with $\delta t^{'} < \delta t_{uv}$ – a contradiction to the choice of $\delta t_{uv} = \min A_{uv}(t)$ as the minimum of $A_{uv}(t)$.

If in contrast, $\not{z}(\delta t_{uv}) = l_{uv}$ holds, let us assume that there exists a $\delta t \in (0, \delta t_{uv})$ such that $\not{z}(\delta t) \in (-l_{uv}, 0)$ holds. Again by the intermediate value theorem, there exists a $\delta t^{'} \in (\delta t, \delta t_{uv})$, such that $\not{z}(\delta t^{'}) = 0$ holds. Consequently, we obtain $\delta t^{'} \in A_{uv}(t)$ with $\delta t^{'} < \delta t_{uv}$ – a contradiction to the choice of $\delta t_{uv} = \min A_{uv}(t)$ as the minimum of $A_{uv}(t)$.

*Step 2.4: For $\delta t_{uv} := \min A_{uv}(t)$, we obtain*

$$\begin{cases} \operatorname{im} z_{|(0,\delta t_{uv})} \subset (-l_{uv}, 0) & \text{if } \not{z}(\delta t_{uv}) = -l_{uv}, \\ \operatorname{im} z_{|(0,\delta t_{uv})} \subset (-l_{uv}, 0) \text{ or } \operatorname{im} z_{|(0,\delta t_{uv})} \subset (0, l_{uv}) & \text{if } \not{z}(\delta t_{uv}) = 0, \\ \operatorname{im} z_{|(0,\delta t_{uv})} \subset (0, l_{uv}) & \text{if } \not{z}(\delta t_{uv}) = l_{uv}. \end{cases}$$

The claim follows immediately from *Step 2.2* and *Step 2.3*.

*Step 2.5: $\delta t_{uv}$ is (always w.r.t. $e_{uv}$ and $t$) ...*

$$\begin{aligned} &\text{... an inverse pass-through time} && \iff \not{z}(\delta t_{uv}) = -l_{uv}, \\ &\text{... a self-loop time or an inverse self-loop time} && \iff \not{z}(\delta t_{uv}) = 0, \\ &\text{... a pass-through time} && \iff \not{z}(\delta t_{uv}) = l_{uv}. \end{aligned}$$

"$\Rightarrow$" If $\delta t_{uv}$ is an inverse pass-through time, by (1) definition of $\not{z}$ and (2) condition (21), we obtain

$$\not{z}(\delta t_{uv}) \overset{(1)}{=} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(2)}{=} -l_{uv}.$$

Analogously, if $\delta t_{uv}$ is a self-loop time or an inverse self-loop time, we use condition (23) or (25) to conclude that $\not{z}(\delta t_{uv}) = 0$ holds. If $\delta t_{uv}$ is a pass-through time, we use condition (19) to conclude that $\not{z}(\delta t_{uv}) = l_{uv}$ holds.

"$\Leftarrow$" If $\not{z}(\delta t_{uv}) = -l_{uv}$ holds, by (1) definition of $\not{z}$ and (2) this assumption, we obtain

$$\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(1)}{=} \not{z}(\delta t_{uv}) \overset{(2)}{=} -l_{uv}.$$

that is, condition (21) is satisfied.

Consequently, for any $t^{'} \in (\min\{t-\delta t_{uv}, t\}, \max\{t-\delta t_{uv}, t\}) = (t-\delta t_{uv}, t)$, or equivalently, for any $t^{'} = t-\delta t$ with $\delta t \in (0, \delta t_{uv})$, by (1) by basic results from analysis, (2) condition (21), (3) definition of $\not{z}$ and (4) *Step 2.4*, we obtain

$$\int_{t-\delta t_{uv}}^{t^{'}} \nu_{uv}(s)\ ds \overset{(1)}{=} \underbrace{\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds}_{\overset{(2)}{=} -l_{uv}} - \underbrace{\int_{t-\delta t}^{t} \nu_{uv}(s)\ ds}_{\overset{(3)}{=} \not{z}(\delta t) \overset{(4)}{\in} (-l_{uv}, 0)} \overset{(1)}{\in} (-l_{uv}, 0),$$

that is, condition (22) is satisfied, too, and thus, $\delta t_{uv}$ is an inverse pass-through time.

Analogously, if $\not{z}(\delta t_{uv}) = 0$ holds, we use this assumption to show that condition (23), which is equal to condition (25), is satisfied. If $\not{z}(\delta t_{uv}) = l_{uv}$ holds, we use this assumption to show that condition (19) is satisfied. Consequently, we use these conditions and *Step 2.4* to show that condition (24), (26) and (20) are satisfied in alignment with the conditions in *Step 2.4*. We leave the details out, but a summary of the cases to be considered is given as follows:

$$\underbrace{\int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds}_{\substack{= -l_{uv} \\ = 0 \\ = 0 \\ = l_{uv}}} - \underbrace{\int_{t-\delta t}^{t} \nu_{uv}(s)\ ds}_{\substack{\in (-l_{uv},0) \\ \in (-l_{uv},0) \\ \in (0,l_{uv}) \\ \in (0,l_{uv})}} \overset{(1)}{=} \underbrace{\int_{t-\delta t_{uv}}^{t'=t-\delta t} \nu_{uv}(s)\ ds}_{\substack{\in (-l_{uv},0) \\ \in (0,l_{uv}) \\ \in (-l_{uv},0) \\ \in (0,l_{uv})}} \quad \begin{array}{l} \text{if } \not{z}(\delta t_{uv}) = -l_{uv} \\ \text{if } \not{z}(\delta t_{uv}) = 0 \text{ and } \operatorname{im} z_{|(0,\delta t_{uv})} \subset (-l_{uv},0) \\ \text{if } \not{z}(\delta t_{uv}) = 0 \text{ and } \operatorname{im} z_{|(0,\delta t_{uv})} \subset (0,l_{uv}) \\ \text{if } \not{z}(\delta t_{uv}) = l_{uv} \end{array}$$

*3.:* Similar to the proof of Lemma C.7.2 and Lemma C.8.1, let us assume that there exist two different and positive times $\delta t_{1uv}, \delta t_{2uv} \in \mathbb{R}_{\neq 0}$ which are each either a pass-through time, an inverse pass-through time, a self-loop time or an inverse self-loop time. W.l.o.g., let $0 < \delta t_{1uv} < \delta t_{2uv}$ hold. Consequently, for $t' = t - \delta t_{1uv} \in (t - \delta t_{2uv}, t) = (\min\{t - \delta t_{2uv}, t\}, \max\{t - \delta t_{2uv}, t\})$, by (1) by basic results from analysis and (2) one or two of the conditions (19), (21), (23) or (25)[20], we obtain

$$\int_{t-\delta t_{2uv}}^{t'} \nu_{uv}(s)\ ds \overset{(1)}{=} \int_{t-\delta t_{2uv}}^{t} \nu_{uv}(s)\ ds - \int_{t-\delta t_{1uv}}^{t} \nu_{uv}(s)\ ds \overset{(2)}{\in} \{-2l_{uv}, -l_{uv}, 0, l_{uv}, 2l_{uv}\}$$

– a contradiction to one of the conditions (20), (22), (24) or (26)[21]. □

### E.11 Proof of Lemma C.15

**Lemma** (Lemma C.15)**.** *Let $e_{uv} \in E$ and $t \in \mathbb{R}$ hold. In the setting of Theorem C.13, the following property holds:*

1. *$\delta t_{uv}$ is (always w.r.t. $e_{uv}$ and $t$) ...*

| | |
|---|---|
| *... a pass-through time* | $\iff \not{z}(\delta t_{uv}) \in \{-l_{uv}, l_{uv}\}$ *and* $q_{uv}(t) > 0$, |
| *... an inverse pass-through time* | $\iff \not{z}(\delta t_{uv}) \in \{-l_{uv}, l_{uv}\}$ *and* $q_{uv}(t) < 0$, |
| *... a self-loop time* | $\iff \not{z}(\delta t_{uv}) = 0$ *and* $q_{uv}(t) < 0$, |
| *... an inverse self-loop time* | $\iff \not{z}(\delta t_{uv}) = 0$ *and* $q_{uv}(t) > 0$. |

*Proof.* "⇒" If $\delta t_{uv}$ is a pass-through time, similarly to *Step 2.5* in the proof of Theorem C.13.2, by (1) definition of $\not{z}$ and (2) condition (19), we obtain

$$\not{z}(\delta t_{uv}) \overset{(1)}{=} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(2)}{=} l_{uv} \in \{-l_{uv}, l_{uv}\}.$$

Moreover, by Lemma C.7 and the choice of $\delta t_{uv} > 0$ , $q_{uv}(t) \geq 0$ holds. Since $q_{uv}(t) \neq 0$ holds by assumption, $q_{uv}(t) > 0$ follows.

[20] Note that we have $3 \cdot 3 = 9$ possible combinations: If $\delta t_{2uv}$ is a pass-through time, we work with condition (19). If $\delta t_{2uv}$ is an inverse pass-through time, we work with condition (21). If $\delta t_{2uv}$ is a(n) (inverse) self-loop time, we work with condition (23), which is equal to condition (25). Each of these three options need to be combined with the three analogous options for the second time $\delta t_{1uv}$.

[21] Note that we have four possible combinations: If $\delta t_{2uv}$ is a pass-through time, we work with condition (20). If $\delta t_{2uv}$ is an inverse pass-through time, we work with condition (22). If $\delta t_{2uv}$ is a self-loop time, we work with condition (24). If $\delta t_{2uv}$ is an inverse self-loop time, we work with condition (26).

Similarly, if $\delta t_{uv}$ is an inverse pass-through time (w.r.t. $e_{uv}$ and $t$), by (1) definition of $\not{z}$ and (2) condition (21), we obtain

$$\not{z}(\delta t_{uv}) \overset{(1)}{=} \int_{t-\delta t_{uv}}^{t} \nu_{uv}(s)\ ds \overset{(2)}{=} -l_{uv} \in \{-l_{uv}, l_{uv}\}.$$

Additionally, we use Lemma C.9 to conclude that $\delta t_{uv}$ is a pass-through time w.r.t. $e_{vu}$ and $t$. Therefore, by (1) the conservation of flows (Eq. (3)), (2) Lemma C.7 and the choice of $\delta t_{uv} > 0$ considered as a pass-through time w.r.t. $e_{vu}$ and $t$, we obtain

$$q_{uv}(t) \overset{(1)}{=} -\underbrace{q_{vu}(t)}_{\overset{(2)}{\geq} 0} \leq 0.$$

Since $q_{uv}(t) \neq 0$ holds by assumption, $q_{uv}(t) < 0$ follows.

Analogously, if $\delta t_{uv}$ is a self-loop time, we use condition (23) to conclude that $\not{z}(\delta t_{uv}) = 0$ holds and Lemma C.8 to conclude that $q_{uv}(t) \leq 0$ and thus, $q_{uv}(t) < 0$ holds. If $\delta t_{uv}$ is an inverse self-loop time, we use condition (25) to conclude that $\not{z}(\delta t_{uv}) = 0$ holds and Lemma C.10, the conservation of flows and Lemma C.8 to conclude that $q_{uv}(t) = -q_{vu}(t) \geq 0$ and thus, $q_{uv}(t) > 0$ holds.

"$\Leftarrow$" If $\not{z}(\delta t_{uv}) \in \{-l_{uv}, l_{uv}\}$ holds, we know by Theorem C.13.2 that $\delta t_{uv}$ is either a pass-through time or an inverse pass-through time.

If additionally, $q_{uv}(t) > 0$ holds, let us assume that $\delta t_{uv}$ is an inverse pass-through time. Consequently, by the same arguments as in the inclusion above, we can conclude that in this case, $q_{uv}(t) = -q_{vu}(t) \leq 0$ holds – a contradiction.

If instead, additionally, $q_{uv}(t) < 0$ holds, let us assume that $\delta t_{uv}$ is a pass-through time. Consequently, by the same arguments as in the inclusion above, we can conclude that in this case, $q_{uv}(t) \geq 0$ holds – a contradiction.

If $\not{z}(\delta t_{uv}) = 0$ holds, we know by Theorem C.13.2 that $\delta t_{uv}$ is either a self-loop time or an inverse self-loop time.

If additionally, $q_{uv}(t) < 0$ holds, let us assume that $\delta t_{uv}$ an inverse self-loop time. Consequently, by the same arguments as in the inclusion above, we can conclude that in this case, $q_{uv}(t) \geq 0$ holds – a contradiction.

If instead, additionally, $q_{uv}(t) > 0$ holds, let us assume that $\delta t_{uv}$ is a self-loop time. Consequently, by the same arguments as in the inclusion above, we can conclude that in this case, $q_{uv}(t) \leq 0$ holds – a contradiction. □

### E.12 Proof of Theorem 3.5 and C.16

**Theorem** (Theorem 3.5 and C.16)**.** *Let $v \in V$ and $t \in \mathbb{R}$ hold. We assume that Assumption 3.4 holds and that for each $u \in \mathcal{N}(v)$, $A_{uv}(t)$ is not empty (that is, the OP (8) has a solution). If the function $c_v$ obeys the mixing at nodes (Eq. (4)) and if for each $u \in \mathcal{N}(v)$, the function $c_{uv}$ obeys the advection edge dynamics (Eq. (2)) with inflow continuity condition (Eq. (5)), then*

$$c_v(t) = \sum_{\substack{u \in \mathcal{N}_-(v,t) \\ \not{z}(\delta t_{uv}) = \pm l_{uv}}} w_{uv}(t) \cdot c_u(t - \delta t_{uv}) + \sum_{\substack{u \in \mathcal{N}_-(v,t) \\ \not{z}(\delta t_{uv}) = 0}} w_{uv}(t) \cdot c_v(t - \delta t_{uv}) = \sum_{u \in \mathcal{N}(v)} w_{uv}(t) \cdot \varphi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu})$$

*holds, where $\delta t_{uv} = \min A_{uv}(t)$ is the transport time as defined in Theorem C.13. In other words, Equation (10) with $\phi$ as defined in Equation (9) holds.*

*Proof.* **Well-definedness:** Since by assumption, for each $u \in \mathcal{N}(v)$, the set $A_{uv}(t)$ as defined in Theorem C.13 is not empty, according to Theorem C.13.1, a transport time $\delta t_{uv} = \min A_{uv}(t) \in (0, \infty)$ exists for each

$u \in \mathcal{N}(v)$. Even more, by Theorem C.13.3, is it uniquely determined by the condition that it is positive, allowing to define the map $\mathcal{N}(v) \to (0, \infty), u \mapsto \delta t_{uv}$ that associates a unique transport time with each neighbor $u \in \mathcal{N}(v)$. Especially to mention, for each $u \in \mathcal{N}(v)$ and by definition of the transport time $\delta t_{uv}$ (Theorem C.13), $\text{ƶ}(\delta t_{uv}) \in \{-l_{uv}, 0, l_{uv}\}$ holds.

**Equality:** By (1) definition of the in-, out- and no-flow neighborhoods and (2) the conservation of flows (Eq. (3)), we obtain

$$\begin{aligned}
\mathcal{N}_{\pm}(v,t) &:\overset{(1)}{=} \{u \in \mathcal{N}(v) \mid \operatorname{sgn}(q_{vu}(t)) = \pm 1\} \overset{(2)}{=} \{u \in \mathcal{N}(v) \mid \operatorname{sgn}(q_{uv}(t)) = \mp 1\} \text{ and} \\
\mathcal{N}_{0}(v) &:\overset{(1)}{=} \{u \in \mathcal{N}(v) \mid \operatorname{sgn}(q_{vu}(t)) = \quad 0\} \overset{(2)}{=} \{u \in \mathcal{N}(v) \mid \operatorname{sgn}(q_{uv})(t) = \quad 0\}.
\end{aligned}$$

Clearly, these three sub-neighborhoods define a partition of the neighborhood $\mathcal{N}(v)$, that is, they are pairwise disjoint and $\mathcal{N}(v) = \mathcal{N}_{-}(v,t) \sqcup \mathcal{N}_{+}(v,t) \sqcup \mathcal{N}_{0}(v,t)$ holds. Together with the definition of the ReLU function $\operatorname{ReLU}(q) = \max\{0, q\}$ for all $q \in \mathbb{R}$, we observe that for each $u \in \mathcal{N}(v)$,

$$\operatorname{ReLU}(q_{uv}(t)) = \begin{cases} q_{uv}(t) & \text{if } u \in \mathcal{N}_{-}(v,t) \\ 0 & \text{if } u \in \mathcal{N}_{+}(v,t) \\ 0 & \text{if } u \in \mathcal{N}_{0}(v,t) \end{cases} \tag{39}$$

holds. Consequently, by (1) Equation (4), (2) Equation (39), (3) the definition of the weights $w_{uv}(t)$ (Eq. (6)), (4) the well-definedness investigations from above and (5) basic transformations, we obtain

$$\begin{aligned}
c_v(t) &\overset{(1)}{=} \frac{\sum_{u \in \mathcal{N}_{-}(v,t)} q_{uv}(t) \cdot c_{uv}(t, l_{uv})}{\sum_{u \in \mathcal{N}_{-}(v,t)} q_{uv}(t)} \\
&\overset{(2)}{=} \frac{\sum_{u \in \mathcal{N}(v)} \operatorname{ReLU}(q_{uv}(t)) \cdot c_{uv}(t, l_{uv})}{\sum_{u \in \mathcal{N}(v)} \operatorname{ReLU}(q_{uv}(t))} \\
&\overset{(3)}{=} \sum_{u \in \mathcal{N}(v)} w_{uv}(t) \cdot c_{uv}(t, l_{uv}) \\
&\overset{(3)}{=} \sum_{u \in \mathcal{N}_{-}(v,t)} w_{uv}(t) \cdot c_{uv}(t, l_{uv}) \\
&\overset{(4,5)}{=} \sum_{\substack{u \in \mathcal{N}_{-}(v,t) \\ \text{ƶ}(\delta t_{uv}) = \pm l_{uv}}} w_{uv}(t) \cdot c_{uv}(t, l_{uv}) + \sum_{\substack{u \in \mathcal{N}_{-}(v,t) \\ \text{ƶ}(\delta t_{uv}) = 0}} w_{uv}(t) \cdot c_{uv}(t, l_{uv}).
\end{aligned}$$

This proves the first equality. Since the definition of $\mathcal{N}_{-}(v,t)$ indicates that $q_{uv}(t) > 0$ is positive, by Lemma C.15.1, the transport times $\delta t_{uv} > 0$ in the first sum are pass-through times, while the transport times $\delta t_{uv} > 0$ in the second sum are inverse self-loop times. Since again by Lemma C.15.1, the other direction also holds, we can conclude the calculations as follows: By (1) applying Theorem C.11 to $c_{uv}(t, l_{uv})$ in the first sum and Lemma C.9 and Theorem C.12 to $c_{uv}(t, l_{uv})$ the second sum, (2) the definition of $\phi$ (Equation (9)), (3) basic transformations, (4) the definition of the transport time $\delta t_{uv}$ (Theorem C.13) and again, (5) the definition of the weights $w_{uv}(t)$ (Eq. (6)), we obtain

$$\begin{aligned}
c_v(t) &\overset{(1)}{=} \sum_{\substack{u \in \mathcal{N}_{-}(v,t) \\ \text{ƶ}(\delta t_{uv}) = \pm l_{uv}}} w_{uv}(t) \cdot c_u(t - \delta t_{uv}) + \sum_{\substack{u \in \mathcal{N}_{-}(v,t) \\ \text{ƶ}(\delta t_{uv}) = 0}} w_{uv}(t) \cdot c_v(t - \delta t_{uv}) \\
&\overset{(2)}{=} \sum_{\substack{u \in \mathcal{N}_{-}(v,t) \\ \text{ƶ}(\delta t_{uv}) = \pm l_{uv}}} w_{uv}(t) \cdot \phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu}) + \sum_{\substack{u \in \mathcal{N}_{-}(v,t) \\ \text{ƶ}(\delta t_{uv}) = 0}} w_{uv}(t) \cdot \phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu}) \\
&\overset{(3)}{=} \sum_{\substack{u \in \mathcal{N}_{-}(v,t) \\ \text{ƶ}(\delta t_{uv}) \in \{-l_{uv}, 0, l_{uv}\}}} w_{uv}(t) \cdot \phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu})
\end{aligned}$$

$$\overset{(4)}{=} \sum_{u \in \mathcal{N}_-(v,t)} w_{uv}(t) \cdot \phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu})$$
$$\overset{(5)}{=} \sum_{u \in \mathcal{N}(v)} w_{uv}(t) \cdot \phi(t, e_{uv}, \mathbf{c}, \boldsymbol{\nu}).$$

This proves the second equality. □

### E.13 Proof of Theorem C.17

**Theorem** (Theorem C.17)**.** *Let $\mathcal{T} = (A, E)$ be a finite tree with root node $a_0 \in A$, **int**ernal nodes $A_{int}$ and **leaf** nodes $A_{leaf}$ (that is, $A = \{a_0\} \sqcup A_{int} \sqcup A_{leaf}$ holds). If each node $a \in A$ is associated with a magnitude $x_a \in \mathbb{R}$ that satisfies*

$$x_a = \sum_{b \in \mathcal{C}(a)} \omega_{ab} \cdot x_b \text{ for all } a \in \{a_0\} \sqcup A_{int}, \tag{40}$$

*where the set $\mathcal{C}(a) \subset A$ consists of the children of $a \in A$ and $w_{ab}$ is a weight of the edge $e_{ab} \in E$ that depends on both the parent $a \in A$ and its child $b \in \mathcal{C}(a)$, then the magnitude $x_0 := x_{a_0}$ of the root node $a_0 \in A$ is determined by the magnitudes $x_{l_p} := x_{a_{l_p}}$ of the leaf nodes $a_{l_p} \in A_{leaf}$ by*

$$x_0 = \sum_{p=(a_0,\ldots,a_{l_p}) \in \mathcal{P}(a_0, A_{leaf})} \omega(a_0, ..., a_{l_p}) \cdot x_{l_p}, \tag{41}$$

*where $\mathcal{P}(a_0, A_{leaf})$ be the set of paths connecting the root node $a_0$ to one of its leaf nodes $A_{leaf}$ and*

$$\omega(a_0, ..., a_{l_p}) := \prod_{l=0}^{l_p - 1} \omega_{a_l a_{l+1}} \tag{42}$$

*is the product of weights along a(n) (arbitrary) path $p = (a_0, ..., a_{l_p}) \in \mathcal{P}$ with length $l_p \in \mathbb{N}$.*

*Proof.* For the sake of brevity, in this proof, we use the abbreviation $\mathcal{P} = \mathcal{P}(a_0, A_{\text{leaf}})$.

*Step 1: For each $l \in \mathbb{N}$, we obtain*

$$x_{a_0} = \sum_{\substack{p=(a_0,\ldots,a_{l_p}) \in \mathcal{P}: \\ l_p \leq l}} \omega(a_0, ..., a_{l_p}) \cdot x_{a_{l_p}} + \sum_{\substack{p=(a_0,\ldots,a_{l_p}) \in \mathcal{P}: \\ l_p > l}} \omega(a_0, ..., a_l) \cdot x_{a_l}. \tag{43}$$

We prove the claim by induction to $l \in \mathbb{N}$.
*Induction base:* If $l = 1$, by (1) Equation (40) applied to $x_{a_0}$, (2) a change in notation, (3) basic transformations, (4) the definition of $\omega$ (cf. (42)) and (5) the choice of $l = 1$, we obtain

$$\begin{aligned}
x_{a_0} &\overset{(1)}{=} \sum_{b \in \mathcal{C}(a_0)} \omega_{a_0 b} \cdot x_b \\
&\overset{(2,3)}{=} \sum_{a_1 \in \mathcal{C}(a_0) \cap A_{\text{leaf}}} \omega_{a_0 a_1} \cdot x_{a_1} + \sum_{a_1 \in \mathcal{C}(a_0) \cap A_{\text{int}}} \omega_{a_0 a_1} \cdot x_{a_1} \\
&\overset{(3)}{=} \sum_{\substack{p=(a_0,a_1) \in \mathcal{P}: \\ l_p = 1}} \omega_{a_0 a_1} \cdot x_{a_1} + \sum_{\substack{p=(a_0,a_1,\ldots,a_{l_p}) \in \mathcal{P}: \\ l_p > 1}} w_{a_0 a_1} \cdot x_{a_1} \\
&\overset{(4)}{=} \sum_{\substack{p=(a_0,a_1) \in \mathcal{P}: \\ l_p = 1}} \omega(a_0, a_1) \cdot x_{a_1} + \sum_{\substack{p=(a_0,\ldots,a_{l_p}) \in \mathcal{P}: \\ l_p > 1}} \omega(a_0, a_1) \cdot x_{a_l}
\end{aligned}$$

$$\overset{(3,5)}{=} \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p\leq l}} \omega(a_0,...,a_{l_p})\cdot x_{a_{l_p}} + \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p> l}} \omega(a_0,...,a_l)\cdot x_{a_l}.$$

*induction hypothesis:* Equation (43) holds for all $\tilde{l} \leq l \in \mathbb{N}$.

*Induction step:* By (1) the *induction hypothesis*, we obtain

$$x_{a_0} \overset{(1)}{=} \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p\leq l}} \omega(a_0,...,a_{l_p})\cdot x_{a_{l_p}} + \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p> l}} \omega(a_0,...,a_l)\cdot x_{a_l}.$$

Similar to the induction base, by (1) Equation (40) applied to $x_{a_l}$, (note that since $l < l_p$, $a_l \in A_{\text{int}}$ must hold), (2) a change in notation, (3) basic transformations and (4) the definition of $\omega$ (cf. (42)), we transform the second summand to

$$\begin{aligned}
&\sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p> l}} \omega(a_0,...,a_l)\cdot x_{a_l}\\
\overset{(1)}{=}& \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p> l}} \omega(a_0,...,a_l)\cdot \left(\sum_{b\in\mathcal{C}(a_l)} \omega_{a_l b}\cdot x_b\right)\\
\overset{(2,3)}{=}& \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p\geq l+1}} \omega(a_0,...,a_l)\cdot \left(\sum_{a_{l+1}\in\mathcal{C}(a_l)\cap A_{\text{leaf}}} \omega_{a_l a_{l+1}}\cdot x_{a_{l+1}} + \sum_{a_{l+1}\in\mathcal{C}(a_l)\cap A_{\text{in}}} \omega_{a_l a_{l+1}}\cdot x_{a_{l+1}}\right)\\
\overset{(3)}{=}& \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p\geq l+1}} \sum_{a_{l+1}\in\mathcal{C}(a_l)\cap A_{\text{leaf}}} \omega(a_0,...,a_l)\cdot \omega_{a_l a_{l+1}}\cdot x_{a_{l+1}}\\
&+ \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p\geq l+1}} \sum_{a_{l+1}\in\mathcal{C}(a_l)\cap A_{\text{in}}} \omega(a_0,...,a_l)\cdot \omega_{a_l a_{l+1}}\cdot x_{a_{l+1}}\\
\overset{(3)}{=}& \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p= l+1}} \omega(a_0,...,a_l)\cdot \omega_{a_l a_{l+1}}\cdot x_{a_{l+1}}\\
&+ \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p> l+1}} \omega(a_0,...,a_l)\cdot \omega_{a_l a_{l+1}}\cdot x_{a_{l+1}}\\
\overset{(4)}{=}& \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p= l+1}} \omega(a_0,...,a_l,a_{l+1})\cdot x_{a_{l+1}} + \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p> l+1}} \omega(a_0,...,a_l,a_{l+1})\cdot x_{a_{l+1}}.
\end{aligned}$$

Bringing it all together, again by (1) basic transformations, we obtain

$$\begin{aligned}
x_{a_0} =& \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p\leq l}} \omega(a_0,...,a_{l_p})\cdot x_{a_{l_p}} + \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p> l}} \omega(a_0,...,a_l)\cdot x_{a_l}\\
=& \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p\leq l}} \omega(a_0,...,a_{l_p})\cdot x_{a_{l_p}}\\
&+ \sum_{\substack{p=(a_0,...,a_{l_p})\in\mathcal{P}:\\ l_p= l+1}} \omega(a_0,...,a_l,a_{l+1})\cdot x_{a_{l+1}}
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{\substack{p=(a_0,\dots,a_{l_p})\in\mathcal{P}:\\ l_p>l+1}} \omega(a_0,\dots,a_l,a_{l+1})\cdot x_{a_{l+1}} \\
&\overset{(1)}{=} \sum_{\substack{p=(a_0,\dots,a_{l_p})\in\mathcal{P}:\\ l_p<l+1}} \omega(a_0,\dots,a_{l_p})\cdot x_{a_{l_p}} \\
&+ \sum_{\substack{p=(a_0,\dots,a_{l_p})\in\mathcal{P}:\\ l_p=l+1}} \omega(a_0,\dots,a_{l_p})\cdot x_{a_{l_p}} \\
&+ \sum_{\substack{p=(a_0,\dots,a_{l_p})\in\mathcal{P}:\\ l_p>l+1}} \omega(a_0,\dots,a_{l+1})\cdot x_{a_{l+1}} \\
&\overset{(1)}{=} \sum_{\substack{p=(a_0,\dots,a_{l_p})\in\mathcal{P}:\\ l_p\leq l+1}} \omega(a_0,\dots,a_{l_p})\cdot x_{a_{l_p}} + \sum_{\substack{p=(a_0,\dots,a_{l_p})\in\mathcal{P}:\\ l_p>l+1}} \omega(a_0,\dots,a_{l+1})\cdot x_{a_{l+1}},
\end{aligned}
$$

which concludes the induction step.

*Step 2: Equation* (41) *holds.*

Since $\mathcal{T}$ is finite, that is, $\mathcal{T}$ an acyclic graph with a finite number of nodes $|A| < \infty$, also each path $p \in \mathcal{P}$ has to be finite. Therefore, we can choose $l = \max_{p\in\mathcal{P}} l_p \in \mathbb{N}$. Consequently, the second sum in *Step 1* is empty and the condition in the first sum in *Step 1* is naturally satisfied. This yields Equation (41). □

### E.14 Proof of Theorem C.20

**Theorem** (Theorem C.20)**.** *Let $v \in V$ and $t \in \mathbb{R}$ hold. Let $\mathcal{T}(v,t)$ be the inflow tree of $v$ at time $t$ as defined in Definition C.18. If $c_v(t)$ is defined by Equation* (10)*, then*

$$
c_v(t) = \sum_{p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)\dots,(u_{l_p},v_{l_p},t_{l_p}))\in\mathcal{P}(v,t)} \left(\prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l)\right)\cdot c_{v_{l_p}}(t_{l_p})
$$

*with weights $w_{u_{l+1}v_l}(t_l)$ according to Equation* (6) *holds.*

*Proof.* For each layer $l \in \mathbb{N}_0$ of the tree, we now associate each node $a_l = (u_l, v_l, t_l) \in A$ of that layer with the magnitude $x_l := x_{a_l} := c_{v_l}(t_l)$.

*Step 1: Together with these magnitudes, $\mathcal{T}(v,t)$ defines a tree in terms of Theorem C.17.*

In order to prove the claim, we first of all investigate on the term $\phi(t_l, e_{u_{l+1}v_l}, \mathbf{c}, \boldsymbol{\nu})$ for a given parent node $a_l = (u_l, v_l, t_l)$ in the $l$-th layer and its child node $a_{l+1} = (u_{l+1}, v_{l+1}, t_{l+1}) \in \mathcal{C}(a_l)$.[22] Indeed, by (1) Equation (9) and (2) the definition of the inflow tree $\mathcal{T}(v,t)$ (Definition C.18), for any $l \in \mathbb{N}_0$, $\phi(t_l, e_{u_{l+1}v_l}, \mathbf{c}, \boldsymbol{\nu})$ is

[22]Especially to mention, by definition of the inflow tree $\mathcal{T}(v,t)$ (Definition C.18), $u_{l+1} \in \mathcal{N}_-(v_t, t_l)$ holds, that is, the edge $e_{u_{l+1}v_l} \in E$ exists.

equal to

$$
\begin{aligned}
&\phi(t_l, e_{u_{l+1}v_l}, \mathbf{c}, \boldsymbol{\nu}) \\
&\overset{(1)}{=} \begin{cases} c_{u_{l+1}}(t_l - \mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})) & \text{if } \mathfrak{z}(\mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})) \in \{-l_{uv}, l_{uv}\}, \\ c_{v_l}(t_l - \mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})) & \text{if } \mathfrak{z}(\mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})) = 0 \end{cases} \\
&\overset{(2)}{=} \begin{cases} c_{u_{l+1}}(t_{l+1}) & \text{if } \mathfrak{z}(\mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})) \in \{-l_{uv}, l_{uv}\}, \\ c_{v_l}(t_{l+1}) & \text{if } \mathfrak{z}(\mathfrak{s}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})) = 0 \end{cases} \\
&\overset{(2)}{=} c_{v_{l+1}}(t_{l+1}) \\
&\overset{(2)}{=} x_{l+1}.
\end{aligned} \tag{44}
$$

Consequently, again by (1) the definition of the inflow tree $\mathcal{T}(v,t)$ (Definition C.18), (2) Equation (10) and (3) Equation (44), for any $a_l \in \{a_0\} \sqcup A_{\text{int}}$, we obtain

$$
\begin{aligned}
x_{a_l} := x_l &\overset{(1)}{=} c_{v_l}(t_l) \\
&\overset{(2)}{=} \sum_{u_{l+1} \in \mathcal{N}_-(v_l, t_l)} w_{u_{l+1}v_l}(t_l) \cdot \phi(t_l, e_{u_{l+1}v_l}, \mathbf{c}, \boldsymbol{\nu}) \\
&\overset{(3)}{=} \sum_{u_{l+1} \in \mathcal{N}_-(v_l, t_l)} w_{u_{l+1}v_l}(t_l) \cdot x_{l+1} \\
&\overset{(1)}{=} \sum_{a_{l+1} \in \mathcal{C}(a_l)} \omega_{a_l a_{l+1}} \cdot x_{a_{l+1}}
\end{aligned}
$$

with weights

$$
\omega_{a_l a_{l+1}} := w_{u_{l+1}v_l}(t_l). \tag{45}
$$

Consequently, $\mathcal{T}(v,t)$ is a tree in terms of Theorem C.17.

*Step 2: Equation* (29) *holds.*

Again by (1) the definition of the inflow tree $\mathcal{T}(v,t)$ (Definition C.18), (2) Theorem C.17 (which we can apply according to *Step 1*) and (3) Equation (45), we obtain

$$
\begin{aligned}
c_v(t) &\overset{(1)}{=} c_{v_0}(t_0) \overset{(1)}{=} x_0 \\
&\overset{(2)}{=} \sum_{p=(a_0,\ldots,a_{l_p}) \in \mathcal{P}(a_0, A_{\text{leaf}})} \omega(a_0, \ldots, a_{l_p}) \cdot x_{l_p} \\
&\overset{(1,3)}{=} \sum_{p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)\ldots,(u_{l_p},v_{l_p},t_{l_p})) \in \mathcal{P}(v,t)} \left( \prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l) \right) \cdot c_{v_{l_p}}(t_{l_p}),
\end{aligned}
$$

which was to show. □

### E.15 Proof of Theorem C.21

**Theorem** (Theorem C.21)**.** *Let $v \in V$ and $t \in T_{hi} \cup T_{fu}$ hold. Let $\hat{\mathcal{T}}(v,t)$ be the interpolation inflow tree of $v$ at time $t$ as defined in Definition C.18. Let $k \geq \max_{p \in \hat{\mathcal{P}}(v,t)} l_p$ hold. If $\hat{c}_v^{(k)}(t)$ is defined by Equation* (11)*, then*

$$
\hat{c}_v^{(k)}(t) = \sum_{p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)\ldots,(u_{l_p},v_{l_p},t_{l_p})) \in \hat{\mathcal{P}}(v,t)} \left( \prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l) \cdot a(t_{l+1}) \right) \cdot \hat{c}_{v_{l_p}}^{(k-l_p)}(t_{l_p})
$$

*with weights* $w_{u_{l+1}v_l}(t_l)$ *and* $a(t_{l+1})$ *according to Equation* (6) *and* (16)*, respectively, holds.*

*Proof.* Similar to the proof of Theorem C.20, for each layer $l \in \mathbb{N}_0$ of the tree, we now associate each node $a_l = (u_l, v_l, t_l) \in A$ of that layer with the magnitude $x_l := x_{a_l} := \hat{c}_{v_l}^{(k-l)}(t_l)$.

*Step 1: Together with these magnitudes,* $\hat{\mathcal{T}}(v,t)$ *defines a tree in terms of Theorem C.17.*

In order to prove the claim, we first of all investigate on the term $\varphi(t_l, e_{u_{l+1}v_l}, \hat{\mathbf{c}}^{(k-(l+1))}, \boldsymbol{\nu})$ for a given parent node $a_l = (u_l, v_l, t_l)$ in the $l$-th layer and its child node $a_{l+1} = (u_{l+1}, v_{l+1}, t_{l+1}) \in \mathcal{C}(a_l)$. Indeed, by (1) Equation (17), for any $l \in \mathbb{N}_0$, $\varphi(t_l, e_{u_{l+1}v_l}, \hat{\mathbf{c}}^{(k-(l+1))}, \boldsymbol{\nu})$ is equal to

$$
\begin{aligned}
&\varphi(t_l, e_{u_{l+1}v_l}, \hat{\mathbf{c}}^{(k-(l+1))}, \boldsymbol{\nu}) \\
&\overset{(1)}{=} \begin{cases} a(t_j) \cdot \hat{c}_{u_{l+1}}^{(k-(l+1))}(t_j) + a(t_{j+1}) \cdot \hat{c}_{u_{l+1}}^{(k-(l+1))}(t_{j+1}) & \text{if } \text{ƶ}(\text{ş}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})) \in \{-l_{uv}, l_{uv}\}, \\ a(t_j) \cdot \hat{c}_{v_l}^{(k-(l+1))}(t_j) + a(t_{j+1}) \cdot \hat{c}_{v_l}^{(k-(l+1))}(t_{j+1}) & \text{if } \text{ƶ}(\text{ş}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})) = 0 \end{cases}
\end{aligned}
$$

for the $j \in I$ for which $t_l - \text{ş}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu}) \in [t_j, t_{j+1})$ holds. Following (1) the definition of the interpolation inflow tree $\hat{\mathcal{T}}(v,t)$ (Definition C.18), we obtain

$$
\begin{aligned}
&\varphi(t_l, e_{u_{l+1}v_l}, \hat{\mathbf{c}}^{(k-(l+1))}, \boldsymbol{\nu}) \\
&\overset{(1)}{=} \begin{cases} \sum\limits_{\substack{t_{l+1} \in \{t_j, t_{j+1}\}: \\ t_l - \text{ş}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu}) \in [t_j, t_{j+1})}} a(t_{l+1}) \cdot \hat{c}_{u_{l+1}}^{(k-(l+1))}(t_{l+1}) & \text{if } \text{ƶ}(\text{ş}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu})) \in \{-l_{uv}, l_{uv}\}, \\ \sum\limits_{\substack{t_{l+1} \in \{t_j, t_{j+1}\}: \\ t_l - \text{ş}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu}) \in [t_j, t_{j+1})}} a(t_{l+1}) \cdot \hat{c}_{v_l}^{(k-(l+1))}(t_{l+1}) & \text{if } \text{ƶ}(\text{ş}(t_l, ee_{u_{l+1}v_l}, \boldsymbol{\nu})) = 0 \end{cases} \\
&\overset{(1)}{=} \sum_{\substack{t_{l+1} \in \{t_j, t_{j+1}\}: \\ t_l - \text{ş}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu}) \in [t_j, t_{j+1})}} a(t_{l+1}) \cdot \hat{c}_{v_{l+1}}^{(k-(l+1))}(t_{l+1}) \\
&\overset{(1)}{=} \sum_{\substack{t_{l+1} \in \{t_j, t_{j+1}\}: \\ t_l - \text{ş}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu}) \in [t_j, t_{j+1})}} a(t_{l+1}) \cdot x_{l+1}.
\end{aligned} \tag{46}
$$

Consequently, again by (1) the definition of the interpolation inflow tree $\mathcal{T}(v,t)$ (Definition C.18), (2) Equation (11), (3) Equation (46) and (4) basic transformations, for any $a_l \in \{a_0\} \sqcup A_{\text{int}}$, we obtain

$$
\begin{aligned}
x_{a_l} := x_l &\overset{(1)}{=} \hat{c}_{v_l}^{(k-l)}(t_l) \\
&\overset{(2)}{=} \sum_{u_{l+1} \in \mathcal{N}_-(v_l, t_l)} w_{u_{l+1}v_l}(t_l) \cdot \varphi(t_l, e_{v_{l+1}v_l}, \hat{\mathbf{c}}^{(k-(l+1))}, \boldsymbol{\nu}) \\
&\overset{(3)}{=} \sum_{u_{l+1} \in \mathcal{N}_-(v_l, t_l)} w_{u_{l+1}v_l}(t_l) \cdot \left( \sum_{\substack{t_{l+1} \in \{t_j, t_{j+1}\}: \\ t_l - \text{ş}(t_l, e_{u_{l+1}v_l}, \boldsymbol{\nu}) \in [t_j, t_{j+1})}} a(t_{l+1}) \cdot x_{l+1} \right) \\
&\overset{(4)}{=} \sum_{u_{l+1} \in \mathcal{N}_-(v_l, t_l)} \sum_{\substack{t_{l+1} \in \{t_j, t_{j+1}\}: \\ t_l - \text{ş}(t_l, e_{(l+1)l}, \boldsymbol{\nu}) \in [t_j, t_{j+1})}} w_{u_{l+1}v_l}(t_l) \cdot a(t_{l+1}) \cdot x_{l+1} \\
&\overset{(1)}{=} \sum_{a_{l+1} \in \mathcal{C}(a_l)} \omega_{a_l a_{l+1}} \cdot x_{a_{l+1}}
\end{aligned}
$$

with weights

$$
\omega_{a_l a_{l+1}} := w_{u_{l+1}v_l}(t_l) \cdot a(t_{l+1}). \tag{47}
$$

Consequently, $\hat{\mathcal{T}}(v,t)$ is a tree in terms of Theorem C.17.

*Step 2: Equation* (30) *holds.*

Again by (1) the definition of the interpolation inflow tree $\hat{\mathcal{T}}(v,t)$ (Definition C.18), (2) Theorem C.17 (which we can apply according to *Step 1*) and (3) Equation (47), we obtain

$$\begin{aligned}
\hat{c}_v^{(k)}(t) &\overset{(1)}{=} \hat{c}_{v_0}^{(k-0)}(t_0) \overset{(1)}{=} x_{a_0} \\
&\overset{(2)}{=} \sum_{p=(a_0,\dots,a_{l_p})\in\mathcal{P}(a_0,A_{\text{leaf}})} \omega(a_0,\dots,a_{l_p})\cdot x_{l_p} \\
&\overset{(1,3)}{=} \sum_{p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)\dots,(u_{l_p},v_{l_p},t_{l_p}))\in\hat{\mathcal{P}}(v,t)} \left(\prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l)\cdot a(t_{l+1})\right)\cdot \hat{c}_{v_{l_p}}^{(k-l_p)}(t_{l_p}),
\end{aligned}$$

which was to show. □

### E.16 Proof of Theorem 3.6

**Theorem** (Theorem 3.6)**.** *There exists a $\kappa\in\mathbb{N}$, such that $\hat{c}_v^{(k)}(t)$ as defined in Equation* (11) *is equal to zero for all $k>\kappa$, $v\in V$ and $t\in T_{hi}\cup T_{fu}$.*

*Proof.* Let $v\in V=V_{\text{b}}\sqcup(V\setminus V_{\text{b}})$ and $t\in T_{\text{hi}}\sqcup T_{\text{fu}}$ hold.

*Case 1:* If $t\in T_{\text{hi}}$ or $t\in T_{\text{fu}}$ and $v\in V_{\text{b}}$ holds, by Equation (11), $\hat{c}_v^{(k)}(t)$ is equal to zero if $k>0$ and equal to $c_v^{(k)}(t)$ if $k=0$.

*Case 2.1:* If $t\in T_{\text{fu}}$, $v\in V\setminus V_{\text{b}}$ and $k\geq\max_{p\in\hat{\mathcal{P}}(v,t)} l_p$ holds, by Theorem C.21, we can express $\hat{c}_v^{(k)}(t)$ as functions of $\hat{c}_{v_{l_p}}^{(k-l_p)}(t_{l_p})$ for paths $p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)\dots,(u_{l_p},v_{l_p},t_{l_p}))\in\hat{\mathcal{P}}(v,t)$. By definition of $\hat{\mathcal{P}}(v,t)$, $t_{l_p}\in T_{\text{hi}}$ or $t_{l_p}\in T_{\text{fu}}$ and $v_{l_p}\in V_{\text{b}}$ holds and even more, $t_{l_p}$ is the first of the times $t_0=t>t_1>\dots>t_{l_p}$ which satisfies this criterion (that is, $t_l\in T_{\text{fu}}$ and $v_l\in V\setminus V_{\text{b}}$ holds for all $l=0,\dots,l_p-1$). Therefore, by Equation (11), for each of these different lengths $l_p\leq k$, $\hat{c}_{v_{l_p}}^{(k-l_p)}(t_{l_p})$ is equal to zero if $k>l_p$ and equal to $c_{v_{l_p}}(t_{l_p})$ for $k=l_p$.

*Case 2.2:* If $t\in T_{\text{fu}}$, $v\in V\setminus V_{\text{b}}$ and $k<\max_{p\in\hat{\mathcal{P}}(v,t)} l_p$ holds, we can only express $\hat{c}_v^{(k)}(t)$ as functions of $\hat{c}_{v_{l_p}}^{(k-l_p)}(t_{l_p})$ for paths $p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)\dots,(u_{l_p},v_{l_p},t_{l_p}))\in\hat{\mathcal{P}}(v,t)$ which additionally satisfy $l_p\leq k$. For these paths, the same findings as in *Case 2.1* hold.

For all remaining paths which satisfy $l_p>k$, we can express $\hat{c}_v^{(k)}(t)$ as functions of $\hat{c}_{v_l}^{(k-l)}(t_l)=\hat{c}_{v_l}^{(0)}(t_l)$ for $l=k<l_p$. Since in this case, $t_l\in T_{\text{fu}}$ and $v_l\in V\setminus V_{\text{b}}$ holds, by Equation (11), $\hat{c}_{v_l}^{(k-l)}(t_l)=\hat{c}_{v_l}^{(0)}(t_l)$ is equal to zero.

*Case 2:* In summary, if $t\in T_{\text{fu}}$ and $v\in V\setminus V_{\text{b}}$ holds, we can express $\hat{c}_v^{(k)}(t)$ as functions of $\hat{c}_{v_{l_p}}^{(k-l_p)}(t_{l_p})$ which are only non-zero for $k\in\{l_p\mid p\in\hat{\mathcal{P}}(v,t)\}$.

Finally, we choose $\kappa=\max_{v\in V}\max_{t\in T_{\text{hi}}\sqcup T_{\text{fu}}}\max_{p\in\hat{\mathcal{P}}(v,t)} l_p$. Then the claim follows as long as $l_p\in\hat{\mathcal{P}}(v,t)$ is finite for all $v\in V$ and $t\in T_{\text{hi}}\sqcup T_{\text{fu}}$. We have discussed that this is the case in Remark C.19. □

### E.17 Proof of Theorem 3.7 and Theorem C.25

**Theorem** (Theorem 3.7 and Theorem C.25)**.** *Let $v\in V$ and $t\in T_{hi}\cup T_{fu}$ hold. Let $\mathcal{T}(v,t)$ and $\hat{\mathcal{T}}(v,t)$ be the inflow tree and the interpolation inflow tree of $v$ at time $t$ as defined in Definition C.18, respectively. Let*

*Assumption C.22 and C.23 hold. Then the prediction error in-between $c_v(t)$ and $\hat{c}_v(t)$ is given by*

$$|c_v(t) - \hat{c}_v(t)|$$
$$\leq \sum_{\tilde{p}\in\hat{\mathcal{P}}(v,t)} a(\hat{t}_1, ..., \hat{t}_{l_p}) \cdot \left( \underbrace{|c_{v_{l_p}}(t_{l_p}) - c_{v_{l_p}}(\hat{t}_{l_p})|}_{\textit{Temporal interpolation error}} + |c_{v_{l_p}}(t_{l_p})| \cdot \underbrace{|w(t_0, ..., t_{l_p-1}) - w(\hat{t}_0, ..., \hat{t}_{l_p-1})|}_{\textit{Accumulated weighting error}} \right)$$

*with joint paths $\tilde{p} = ((v_0, t_0) = (v, t), (u_1, v_1, t_1, \hat{t}_1), ..., (u_{l_p}, v_{l_p}, t_{l_p}, \hat{t}_{l_p})) \in \hat{\mathcal{P}}(v, t)$ according to Definition C.24,*

$$a(\hat{t}_1, ..., \hat{t}_{l_p}) := \prod_{l=0}^{l_p-1} a(\hat{t}_{l+1}) \quad \textit{and} \quad w(\hat{t}_0, ..., \hat{t}_{l_p-1}) := \prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(\hat{t}_l)$$

*with weights $w_{u_{l+1}v_l}(t_l)$ and $a(t_{l+1})$ according to Equation (6) and (16), respectively. If the temporal interpolation error is zero, we obtain $|c_v(t) - \hat{c}_v(t)| = 0$.*

*Proof.* By Theorem C.20 and C.21,

$$c_v(t) = \sum_{p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)...,(u_{l_p},v_{l_p},t_{l_p}))\in\mathcal{P}(v,t)} \left( \prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l) \right) \cdot c_{v_{l_p}}(t_{l_p}) \text{ and}$$
$$\hat{c}_v^{(k)}(t) = \sum_{p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)...,(u_{l_p},v_{l_p},t_{l_p}))\in\hat{\mathcal{P}}(v,t)} \left( \prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l) \cdot a(t_{l+1}) \right) \cdot \hat{c}_{v_{l_p}}^{(k-l_p)}(t_{l_p})$$

holds for all $k \geq \max_{p\in\hat{\mathcal{P}}(v,t)} l_p$. As already discussed in the proof of Theorem 3.6, for a $k < \max_{p\in\hat{\mathcal{P}}(v,t)} l_p$, paths $p \in \hat{\mathcal{P}}(v, t)$ with $l_p \leq k$ contribute with the same contribution $\hat{c}_{v_{l_p}}^{(k-l_p)}(t_{l_p}) = c_{v_{l_p}}(t_{l_p})$ to $\hat{c}_v^{(k)}(t)$ as they would for a larger $k$, while the ones with $l_p > k$ contribute with $\hat{c}_{v_l}^{(k-l)}(t_l) = 0$ (for $l = k < l_p$). More specifically, considering a flexible $k \in \mathbb{N}$, each path contributes only once to $\hat{c}_v^{(k)}(t)$, namely for $k = l_p$. Therefore, $\hat{c}_{v_{l_p}}^{(k-l_p)}(t_{l_p}) = \delta_{kl_p} \cdot c_{v_{l_p}}(t_{l_p})$ holds for all $k \geq l_p$ and also gives a meaning for cases $k < l_p$. Consequently, by (1) Equation (11) and (2) the findings above, we can express $\hat{c}_v(t)$ by

$$\hat{c}_v(t) \overset{(1)}{=} \sum_{k=0}^{\infty} \hat{c}_v^{(k)}(t)$$
$$\overset{(2)}{=} \sum_{k=0}^{\infty} \sum_{p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)...,(u_{l_p},v_{l_p},t_{l_p}))\in\hat{\mathcal{P}}(v,t)} \left( \prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l) \cdot a(t_{l+1}) \right) \cdot \hat{c}_{v_{l_p}}^{(k-l_p)}(t_{l_p})$$
$$\overset{(2)}{=} \sum_{p=((v_0,t_0)=(v,t),(u_1,v_1,t_1)...,(u_{l_p},v_{l_p},t_{l_p}))\in\hat{\mathcal{P}}(v,t)} \left( \prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l) \cdot a(t_{l+1}) \right) \cdot c_{v_{l_p}}(t_{l_p}).$$

Now we want to use the shared paths $\tilde{p}$ from Equation (31) to express $c_v(t)$ and $\hat{c}_v(t)$ using the same summands. To achieve this, it is important to note that as an artifact of the interpolation, there are multiple paths $\hat{p} \in \hat{\mathcal{P}}(v, t)$ that are identifiable with a single path $p \in \mathcal{P}(v, t)$. However, due to the fact that certain interpolation weights $a(t_{l+1})$ add up to one, a single contribution $c_{v_{l_p}}(t_{l_p})$ of a path $p \in \mathcal{P}(v, t)$ to $c_v(t)$, weighted by the weight

$$\left( \prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l) \right),$$

can be distributed to the multiple contributions $c_{v_{l_p}}(t_{l_p})$ of a path $p \in \mathcal{P}(v,t)$ to $c_v(t)$, weighted by the weights

$$\left(\prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l) \cdot a(t_{l+1})\right)$$

associated with corresponding paths $\hat{p} \in \hat{\mathcal{P}}(v,t)$ that each are identifiable with the path $p \in \mathcal{P}(v,t)$. Using the shared paths $\tilde{p}$ notation from Equation (31), this allows to express both $c_v(t)$ and $\hat{c}_v(t)$ as

$$\begin{aligned}
c_v(t) &= \sum_{\tilde{p}=((v_0,t_0)=(v,t),(u_1,v_1,t_1,\hat{t}_1),\ldots,(u_{l_p},v_{l_p},t_{l_p},\hat{t}_{l_p}))\in\hat{\mathcal{P}}(v,t)} \left(\prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l) \cdot a(\hat{t}_{l+1})\right) \cdot c_{v_{l_p}}(t_{l_p}) \text{ and} \\
\hat{c}_v^{(k)}(t) &= \sum_{\tilde{p}=((v_0,t_0)=(v,t),(u_1,v_1,t_1,\hat{t}_1),\ldots,(u_{l_p},v_{l_p},t_{l_p},\hat{t}_{l_p}))\in\hat{\mathcal{P}}(v,t)} \left(\prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(\hat{t}_l) \cdot a(\hat{t}_{l+1})\right) \cdot c_{v_{l_p}}(\hat{t}_{l_p}),
\end{aligned}$$

making the sums perfectly comparable. Therefore, by (1) basic transformations and triangle-inequality and (2) the fact that $|w_1c_1 - w_2c_2| = |w_1c_1 + w_2c_1 - w_2c_1 - w_2c_2| = |w_2c_1 - w_2c_2 + c_1w_1 - c_1w_2| \leq |c_1 - c_2| + |c_1||w_1 - w_2|$ holds for any $w_1, w_2 \in [0,1]$ and $c_1, c_2 \in \mathbb{R}$, we obtain

$$\begin{aligned}
&|c_v(t) - \hat{c}_v(t)| \\
\overset{(1)}{\leq} &\sum_{\tilde{p}\in\hat{\mathcal{P}}(v,t)} \left(\prod_{l=0}^{l_p-1} a(\hat{t}_{l+1})\right) \cdot \left| \left(\prod_{l=0}^{l_p-1} \underbrace{w_{u_{l+1}v_l}(t_l)}_{\in[0,1]}\right) \cdot c_{v_{l_p}}(t_{l_p}) - \left(\prod_{l=0}^{l_p-1} \underbrace{w_{u_{l+1}v_l}(\hat{t}_l)}_{\in[0,1]}\right) \cdot c_{v_{l_p}}(\hat{t}_{l_p}) \right| \\
\overset{(2)}{\leq} &\sum_{\tilde{p}\in\hat{\mathcal{P}}(v,t)} \left(\prod_{l=0}^{l_p-1} a(\hat{t}_{l+1})\right) \cdot \Bigg( \quad |c_{v_{l_p}}(t_{l_p}) - c_{v_{l_p}}(\hat{t}_{l_p})| \\
&\qquad + \quad |c_{v_{l_p}}(t_{l_p})| \cdot \left| \left(\prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(t_l)\right) - \left(\prod_{l=0}^{l_p-1} w_{u_{l+1}v_l}(\hat{t}_l)\right) \right| \Bigg).
\end{aligned}$$

If the temporal interpolation error is zero, that is, if $\phi = \varphi$ holds, all interpolation times $\hat{t}_l$ corresponding to left interpolation bound will be equal to $t_l$ hand have interpolation weights $a(\hat{t}_l) = 1$. Interpolation times $\hat{t}_l$ corresponding to right interpolation bounds have interpolation weights $a(\hat{t}_l) = 0$ (cf. Eq. (16). Therefore, in this case, the only non-zero summands in the error bound are the ones where $t_l = \hat{t}_l$ holds for all $l = 0, ..., l_p$, in which case $|c_v(t) - \hat{c}_v(t)| = 0$ holds. This proves the claim.

□